AI HEALTH MONITORING AMBU VENTILATOR

MINI PROJECT REPORT

Submitted by

SRIDHAR.A	**(612720114021)**
VISWANATHAN.S	**(612720114028)**
AJITH.S	**(612720114301)**
JAISURYA.S	**(612720114007)**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

MECHANICAL ENGINEERING

THE KAVERY ENGINEERING COLLEGE

SALEM-636453

ANNA UNIVERSITY::CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified the project report **"AI HEALTH MONITORING AMBU VENTILATOR"** is the bonafide work of **"SRIDHAR.A (612720114021), VISWANATHAN.S (612720114028), AJITH.S (612720114301), JAISURYA.S (612720114007)"** Who carried out the Project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.BENJAMIN LAZARUS, M.E., Ph.D.,	**Mr.M.RAMAMOORTHY, M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor,	Assistant professor,
Dept. of Mechanical Engineering,	Dept. of Mechanical Engineering,
The Kavery Engineering College,	The Kavery Engineering College,
Mecheri, Salem-636 453.	Mecheri, Salem-636 453.

Submitted for the project viva voce held on ___________________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We thank the Almighty, who has given his grace and wisdom to complete our project work successfully. We wish to convey our sincere gratitude to our respected Honorable Chairman and Advisor Prof. **Dr. A. K. NATESAN** and the Vice Chairman **Mr. A. ANBALAGAN** for the facilities provide to complete this project. We express our whole hearted gratitude to our respected secretary **Prof. S. K. ELANGOVAN** and respected coordinator **Mrs. S. V. REVATHI ELANGOVAN** for their inspiring support.

We would like to express our sincere thanks to our respected Principal **Dr. V. DURAISAMY** for allowing us to do our project and offering adequate duration to complete it. We are grateful to the Head of the Department **Dr. S.BENJAMIN LAZARUS** for his valuable suggestions and encouragement during our project phase.

We would like to express our deep sense of gratitude and heartfelt thanks to our guide **Mr. M.RAMAMOORTHY** Assistant Professor in Department of Mechanical Engineering, right from deciding the topic, finalizing the thesis work, method of presentation and making necessary corrections. We are very thankful for her valuable guidance, constant encouragement and keen involvement during this project work.

We also extend our indent thanks our teaching staffs and non- teaching staffs of Mechanical Engineering Department for rending their valuable support. And finally it is with a profound feeling of joy that we convey our heartfelt thanks to all our friends for their constant encouragement during the course of our project.

ABSTRACT

Due to contraction motion in the diaphragm, reverse pressure is produced . By using this reverse pressure, human lungs suck the air from the environment for breathing.Recent days the pulmonary related problems are increasing. The healthcare system does not provide a proper ventilators to patients affected by respiratory disease. The technology is developing day by day. The lack of ventilators wasa struggle for every country around the world. One of the leading factors for the shortageof ventilators was due to their cost. This situation affected developing and under- developed countries the most. The existing ventilator technology is completely based on IOT which requires a manual support. The main work done in this thesis is towards the development, data collection, analysis and validation of the controller used for a low-cost mechanical ventilator device designed using a ambu bag valve mask (BVM) and a microcontroller. Our project is done to overcome this by using the AI technology, which operates both manual and artificial support.. Using AI ventilator with speed control and calculate the working hours of ventilator. Ventilator motor is controlled by a simplified computer device.MG995 servo motor with and speed displayed in lcd display .one side push mechanism to push the ventilator ambu bag .MAX30100 heart rate sensor ,temperature sensor,ESP8266 Development board with wifi has been utilized. Here using this ventilator heart rate and spO2 level,temperature ambu bag pressure ,volume controlled in real time and displayed in lcd display and android smartphones.

TABLE OF CONTENT

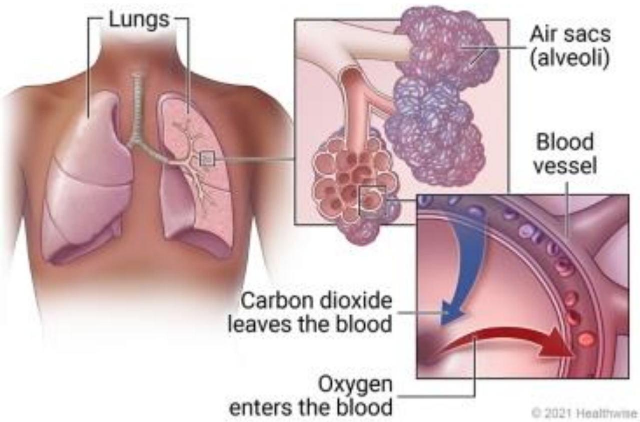
CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The ultimate aim of our project is to monitor spO2 level,body temperature ,heart beats and also recover the patients who is affected respiratory problems using artificial intelligence based ventilator.

1.2 LUNGS ANATOMY



The lungs are the foundational organs of the respiratory system, whose most basic function is to facilitate gas exchange from the environment into the bloodstream. Oxygen gets transported through the alveoli into the capillary network, where it can enter the arterial system, ultimately to perfuse tissue. The lung has an apex, three borders, and three surfaces. The apex lies above the firstrib. The three borders include the anterior, posterior, and inferior borders. The anterior border of the lung corresponds to the pleural reflection, and it creates a cardiac notch in the left lung.

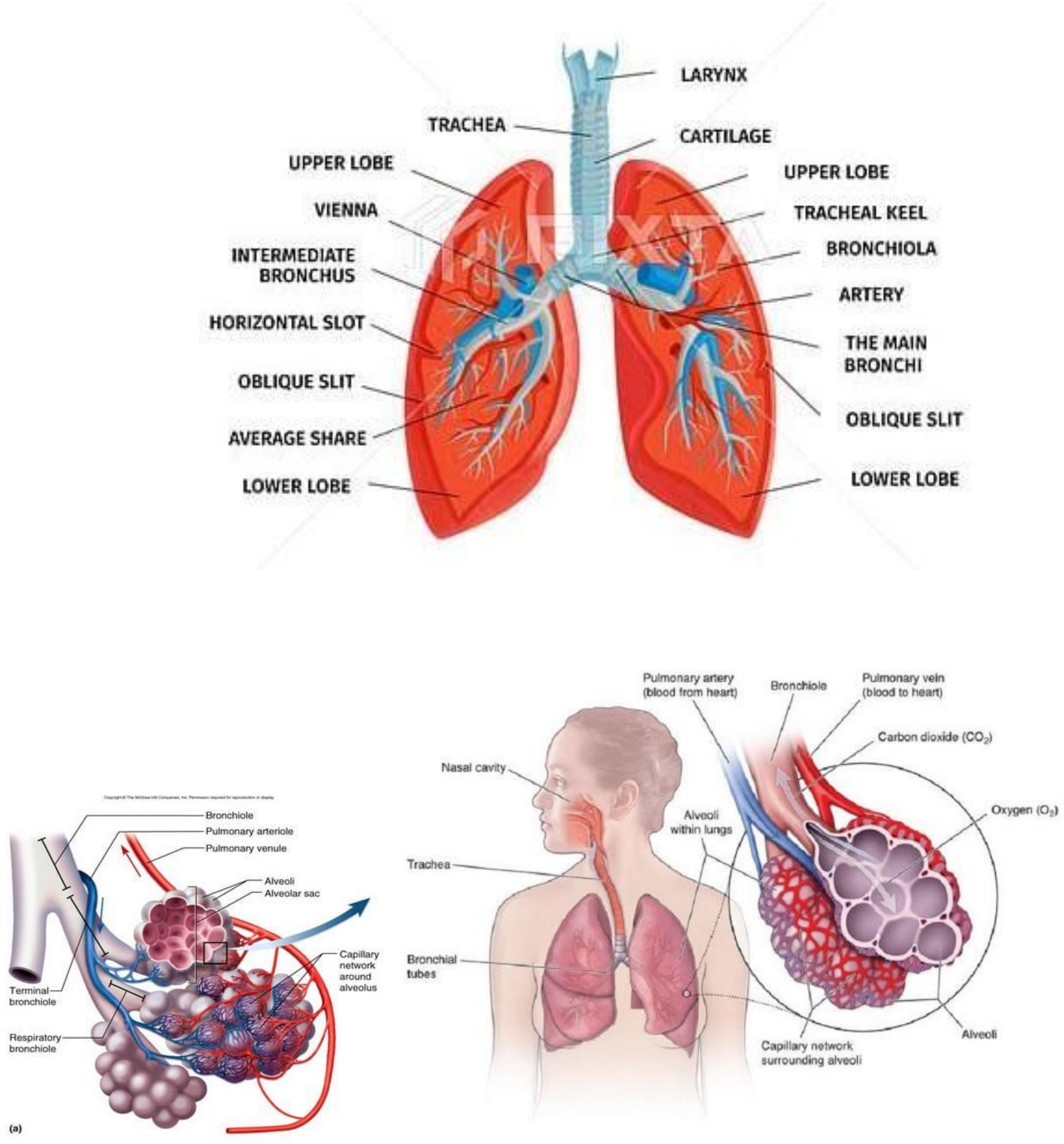


Fig 1.1 Structure of lungs

The breathing rate varies from person to person, ranging from 15-18 times per minute. When it comes to understanding the mechanism of respiration. The pressure of air plays a major role. The alveolar spaces of the lungs take in air by the active process of inspiration, when the pressure inside the lungs exceeds the pressure of the atmosphere, the oxygen comes out as carbon dioxide during expiration, is known as mechanism of breathing.

1.3 VENTILATOR

Ventilators are machines that act as bellows to move air in and out of our lungs. Our respiratory therapist and doctor set the ventilator to control how often it pushes air into your lungs and how much air you get. The ventilator fitted with a mask to get air into our lungs. All living beings need oxygen to live. The process of breaking down food particles into energy also requires an oxygen supply. Saying this, breathing is a respiratory process of taking in oxygen and exhaling carbon dioxide. This is also called ventilation. In short, the out and in movementsof air into the lungs is called breathing or external ventilation.

1.3.1 PARAMETERS OF RESPIRATION

The parameters of respiration are measurements that indicate the state of respiratory function, including lung volumes and capacities, airway resistance, lung compliance, elasticity and intrathoracic pressure. The volumes are tidal volume (500ml), inspiratory reserve volume (3600ml), expiratory reserve volume (1200ml) and residual volume (1200ml). The capacities are vital capacity (4800ml), inspiratory capacity (3600ml), functional residual capacity (2400ml) andtotal lung capacity (6000ml).

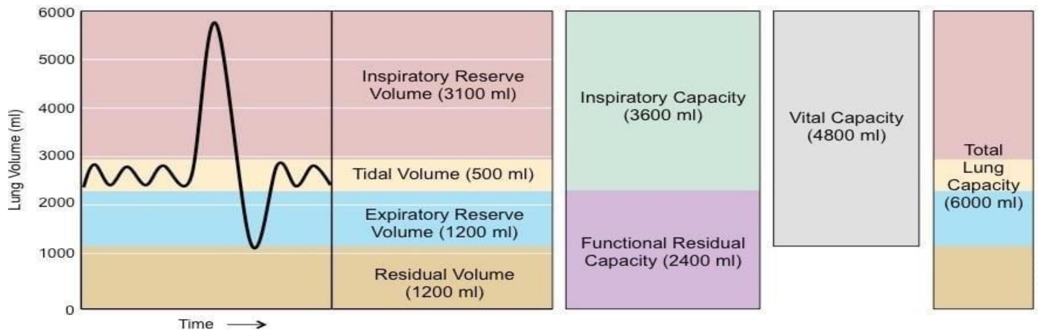


Fig 1.2 Respiratory volumes and capacities

1.4 PULMONARY DISEASES

A group of lung diseases that block airflow and make it difficult to breathe. Pulmonary disease is any problem in the lungs that prevents the lungs from working properly. It main causes, exposure to lung irritants like tobacco smoke ,Radon,asbestos,or other form of air pollution

There are three main types of lung disease:

1.4.1 Airway diseases

These diseases affect the tubes (airways) that carry oxygen and other gases into and out of the lungs. They usually cause a narrowing or blockage of the airways. Airway diseases include asthma, chronic obstructive pulmonary disease (COPD), bronchiolitis, and bronchiectasis (which also is the main disorder for persons with cystic fibrosis). People with airway diseases often say they feel as if they're "trying to breathe out through a straw."

1.4.2 Lung tissue diseases

These diseases affect the structure of the lung tissue. Scarring or inflammation of the tissue makes the lungs unable to expand fully (restrictive lung disease). This makes it hard for the lungs to take in oxygen and release carbon dioxide. People with this type of lung disorder often say they feel as if they are "wearing a too-tight sweater or vest." As a result, they can't breathe deeply. Pulmonary fibrosis and sarcoidosis are examples of lung tissue disease.

1.4.3 Lung circulation diseases

These diseases affect the blood vessels in the lungs. They are caused by clotting, scarring, or inflammation of the blood vessels. They affect the ability of the lungs to take up oxygen and release carbon dioxide. These diseases may also affect heart function. An example of a lung circulation disease is pulmonaryhypertension. People with these conditions often feel very short of breath when they exert themselves.

1.5 ARTIFICIAL INTELLIGENCE

The ventilator is made up by the artificial intelligence technique. Artificial intelligence (AI) is intelligence perceiving, synthesizing, and inferring information demonstrated by machines, as opposed to intelligence displayed by non-human animals or by humans. Example tasks in which this is done include speech recognition, computer vision, translation between (natural) languages, as well as other mappings of inputs. Applications include advanced web search engines (e.g., Google Search), recommendation systems, understanding human speech, self-driving cars, generative or creative tools, automated decision-making, and competing at the highest level in strategic game systems.

The various sub-fields of AI research are centered on particular goals and the use of particular tools. The traditional goals of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception, and the ability to move and manipulate objects. General intelligence (the ability to solve an arbitrary problem) is among the field's long-term goals.

CHAPTER 2

LITERATURE REVIEW

2.1 DESIGNING AND ELECRO MECHANICAL VENTILATOR BASEDON DOUBLE CAM INTEGRATION MECHANISM

This paper proposes a simplified structure of microcontroller based mechanical ventilator integrated with a Bag-Valve Mask (BVM) ventilation mechanism. Here, an Ambu bag is operated with computer-aided manufacturing (CAM) arm that is commanded via a microcontroller and manual switches by sending a control signal to the mechanical system and according to this control signal, the mechanical computer-aided manufacturing (CAM) arm simultaneously compresses and decompresses the Ambu bag. It is a self-inflating bag and like a one-way valve around its inlet and outlet corner. By compressing the Ambu bag it delivers air and by relaxing, it takes air from the environment through a mechanical scavenger. The control signals are designed with three modes named adult mode, pediatric mode, and child mode based on the respiratory rate. The device is in assist controlled mode by dint of fixing the tidal volume for all unique control signals. The control signal is visualized by a platform known as the BIOPAC student's lab system. The proposed device is portable, compact, low weight, and efficient performable. It can be supplied around the rural area hospitals for immediate medication with cost efficiency and risk avoidance. Anyone can operate it as no need to study or training of ventilation rules like ICU ventilator. The proposed system is safe, riskless, and repairable. The angle, volume, and respiratory measurement have found 95%, 92%, and 90% accuracy respectively. By applying this portable ventilator system immediate attention can be taken up in rural or general hospitals and in ambulances.

2.2 CLOSED LOOP CONTROL OF MECHANICAL VENTILATION OFTARGETING SCHEMES, RESPIRATORY CARE

There has been a dramatic increase in the number and complexity of new ventilation modes over the last 30 years. The impetus for this has been the desire to improve the safety, efficiency, and synchrony of ventilator-patient interaction. Unfortunately, the proliferation of names for ventilation modes has made understanding mode capabilities problematic. New models are generally based on increasingly sophisticated closed-loop control systems or targeting schemes. We describe the 6 basic targeting schemes used in commercially available ventilators today: set-point, dual, servo, adaptive, optimal, and intelligent. These control systems are designed to serve the 3 primary goals of mechanical ventilation: safety, comfort, and liberation. The basic operations of these schemes may be understood by clinicians without any engineering background, and they provide the basis for understanding the wide variety of ventilation modes and their relative advantages for improving patient - ventilator synchrony. Conversely, their descriptions may provide engineers with a means to better communicate to end users.

2.3 DESIGN AND PROTOTYPING OF A LOW COST PORTABLE MECHANICAL VENTILATER

This paper describes the design and prototyping of a low-cost portable mechanical ventilator for use in mass casualty cases and resource-poor environments. The ventilator delivers breaths by compressing a conventional bag-valve mask (BVM) with a pivoting cam arm, eliminating the need for a human operator for the BVM. An initial prototype was built out of acrylic, measuring 11.25 x 6.7 x 8 inches (285 x 170 x 200 mm) and weighing 9 lbs. Tidal volume and number of breaths per minute are set via user-friendly input knobs. The prototype also features an assist-control mode and an alarm to indicate over-pressurization of the system. Through this prototype, the strategy of cam-actuated BVM compression is proven to be a viable option to achieve low cost, low-power portable ventilator technology that provides essential ventilator features at a fraction of the cost of existing technology.

2.4 PARTIALLY REPRAPABLE AUTOMATED OPEN SOURCE BAG VALVE MASK BASED VENTILATOR

This study describes the development of a simple and easy-to-build portable automated bag valve mask (BVM) compression system, which, during acute shortages and supply chain disruptions can serve as a temporary emergency ventilator. The resuscitation system is based on the Arduino controller with a real-time operating system installed on a largely RepRap 3-D printable parametric component-based structure. The cost of the materials for the system is under $170, which makes it affordable for replication by makers around the world. The system is designed for reliability and scalability of measurement circuits through the use of the serial peripheral interface and has the ability to connect additional hardware due to the object-oriented algorithmic approach. Experimental results after testing on an artificial lung for peak inspiratory pressure (PIP), respiratory rate (RR), positive end-expiratory pressure (PEEP), tidal volume, proximal pressure, and lung pressure demonstrate repeatability and accuracy exceeding human capabilities in BVM-based manual ventilation. Future work is necessary to further develop and test the system to make it acceptable for deployment outside of emergencies suchas with COVID-19 pandemic in clinical environments, however, the nature of the design is such that desired features are relatively easy to add using protocols and parametric design files provided.

2.5 IOT ENABLED VENTILATOR MONITORING SYSTEM FORCOVID -19 PATIENTS

The use of the IOT protocol on medical equipment is expected to provide protection for medical personnel in dealing with Covid-19 patients, especially when medical personnel are monitoring and setting up an equipment. This study aims to produce a monitoring and control system for a breathing apparatus (Ventilator) based on the Internet of Thing (IOT), test the ventilator control function, test the data transmission function with the IOT protocol. Data collection is done through Testing and Observation, Limited field test. This research producesa control and monitoring system for mechanical ventilators. The mechanical ventilator consists of a gripper motion mechanism driven by a dc motor. The movement of the gripper creates pressure and releases pressure on the ambu bag. The depth of pressure exerted by the gripper is measured as the volume and pressure of the air delivered to the lungs. The rate of pressure exerted is measured as the velocity of air flowing into the lungs.

CHAPTER 3

SYSTEM ANALYSIS

This system analysis provides a background into mechanical ventilation, and a discussion about key ventilator settings and waveforms for reading and analyzing output signals. A literature search was conducted using Google Scholar for literature published in the range from 2010 to 2021. The topics covered in this chapter include breathing anatomy under which mechanism of breathing, respiratory modes and factors that affect breathing are discussed. Mechanical ventilators, including types, working mechanisms, ventilator settings, uses, and risks.

3.1 EXISITING SYSTEM

3.1.1 LOW COST PORTABLE VENTILATOR

To overcome the shortage of ventilators in medical infrastructure, various low-cost, easy to assemble, portable ventilators have been proposed to fight the ongoing pandemic. These mechanical ventilators are made from components that are generally readily available worldwide. Such components are already associated with day-to-day gadgets or items and which do not require specialized manufacturing processes. Various designs have been proposed, focusing on meeting basic requirements for artificial ventilation to fight the ongoing pandemic. This article aims to provide readers an overview of various design parameters that needs to be considered while designing portable ventilators, by systematic analysis from available pool of proposed designs.

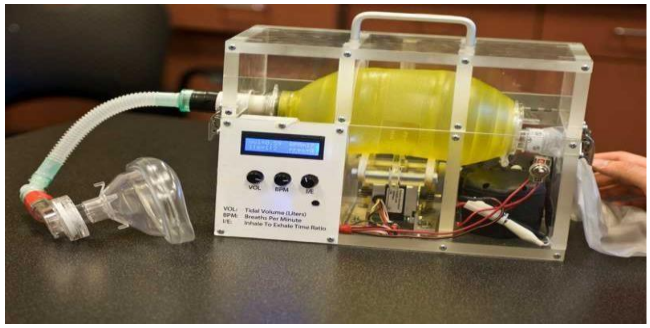


Fig 3.1 Low Cost Portable Ventilator

By going through existing literature, we have recognized multiple factors influencing device performance and how these factors need to be considered for efficient device operation. This article aims to provide readers an overview of various design parameters that needs to be considered while designing portable ventilators, by systematic analysis from available pool of proposed designs.

3.1.2 BAG VALVE MASK (BVM) VENTILATOR

A bag valve mask, sometimes known by the proprietary name ambu bag or generically as a manual resuscitator or "self-inflating bag", is a hand-held device commonly used to provide positive pressure ventilation to patients who are not breathing or not breathing adequately.

Fig 3.2 Bag valve mask ventilator

The mask is manually held tightly against the face, and squeezing the bag ventilates the patient through the nose and mouth. Unless contraindicated, airway adjuncts such as nasopharyngeal and/or oropharyngeal airways are used during BVM ventilation to assist in creating a patent airway.

3.1.3 IOT BASED VENTILATOR

It is essential to develop low-cost and high-efficiency technologies for real-time health detection and continuous treatment system. The IOT - based ventilator is a low-cost, respiratory monitoring, and controlling system. The designed and developed ventilator delivers breaths by compressing a conventional ambu bag with the help of a fixed arm connected to a servo motor. It eliminates theneed for a human operator for the bag valve mask. Tidal volume and number of breaths per minute are set via user-friendly input modes, which help in customizingthe pressure according to the real-time requirements of the patient. It has a safety mode that is when the power supply goes down; the backup battery automatically kicks in. A built-in alerts system is embedded to activate and warn the local andthe remote locations about any malfunctioning for immediate attention.

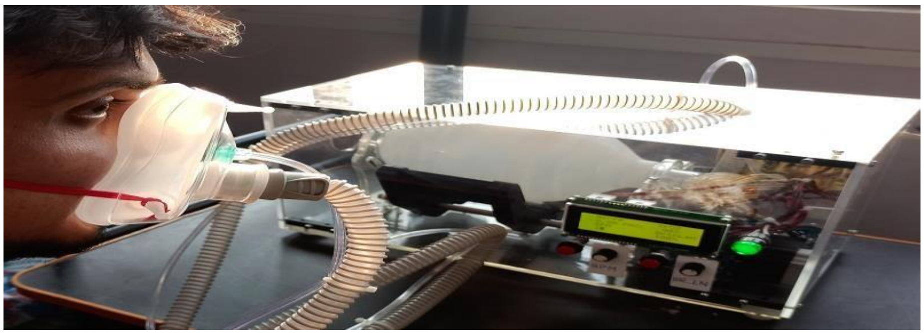


Fig 3.3 IOT based Ventilator

3.2 PROPOSED SYSTEM

In this system to control the parameters of compressed air and oxygen mixture like flow, pressure, the fraction of inspired oxygen (FiO2), respiration rate, I: E ratio, etc., components like a pressure regulator, proportional valve, pressure sensors, flow sensors, etc. are used. The functioning of these components is monitored and controlled by the control unit to perform mechanical ventilation properly. The actual design of a mechanical ventilator is too complicated and thus it is very costly. It is a completely controlled Both manual and mobile application called blynk.The motor speed can be controlled at anytime so we can control the oxygen flow in to human body. Thus increasing the mechanical speed of motor the mechanical arm connected to air bag keep pushing which produces continuous air flow .The air flow is fed to the mask which is given to the patient.

3.2.1 CONTRIBUTION

- An Arduino Due micro-controller device was programmed to control the system. The user inputs TV, BPM and I/E ratio. These input data determine the speed of the motor was controlled by remote.
- Using AI software the spo2 rate was properly controlled. The working hours of the AI based ventilator was calculated, it is more useful for ventilator manufacturing companies.
- The device was evaluated for various resistance of the lungs through and artificial lung device. LED Light for representing patient condition.

CHAPTER 4

HARDWARE

4.1 FUNCTIONAL BLOCK DIAGRAM

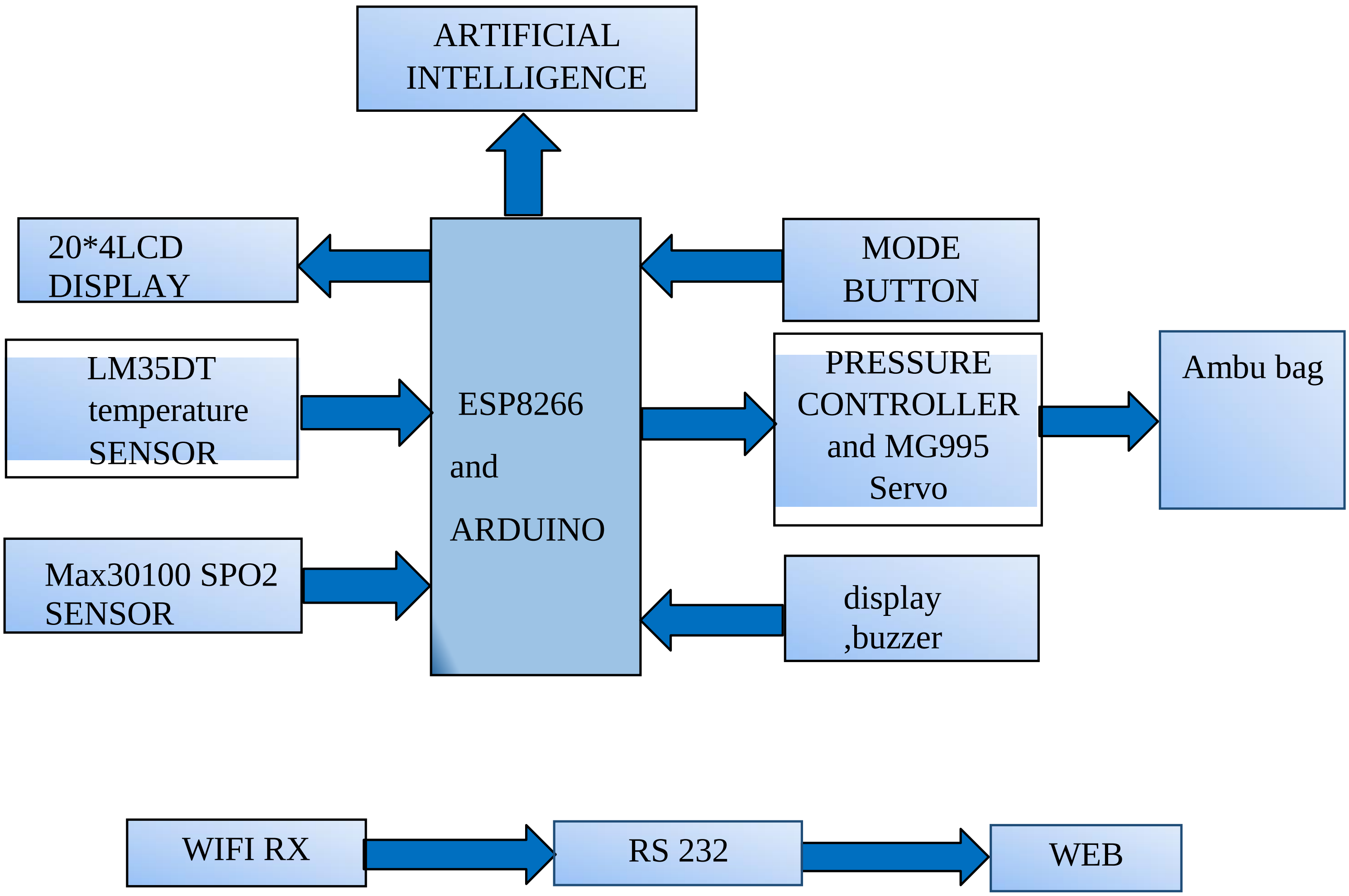
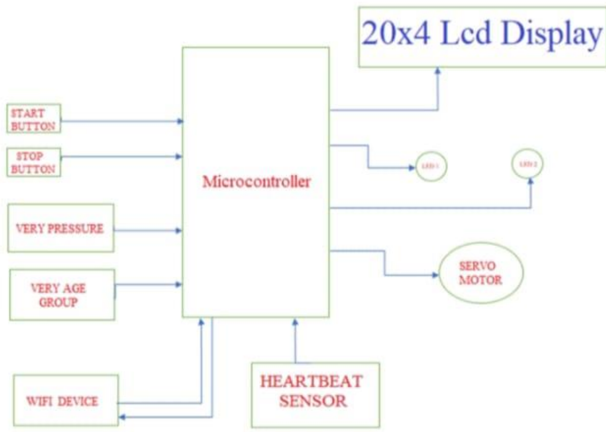


Fig 4.1 Functional Block Diagram

Ventilators gently force normal air or a mixture of air and added oxygen through a breathing tube, into a patient's airways and down into their lungs.

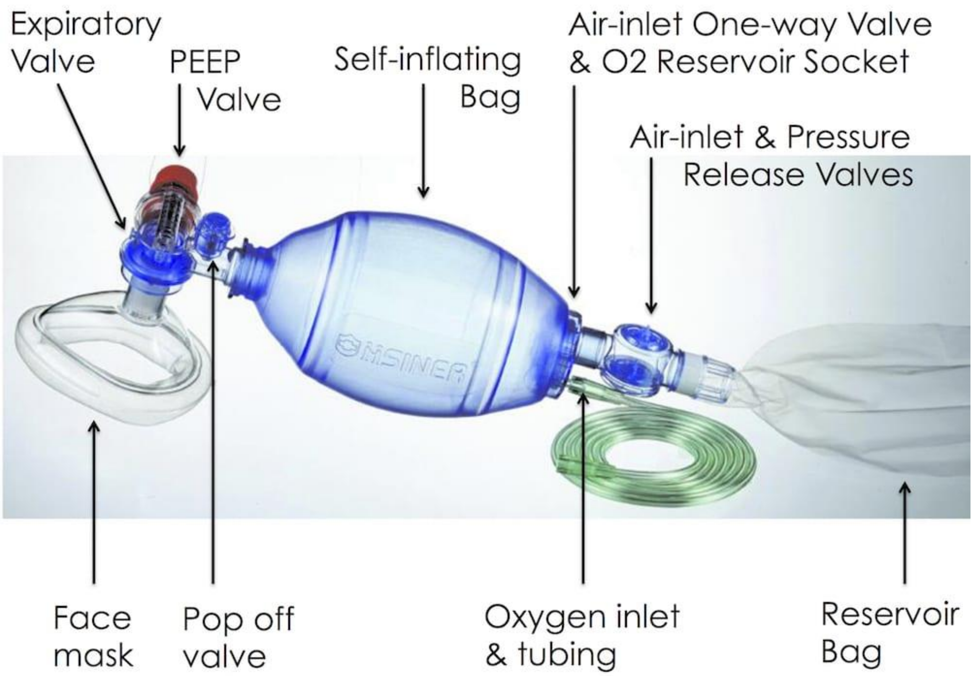


The ventilator mechanism must be able to deliver in the range of 10 – 30 breaths per minute, with the ability to adjust rising increments in sets of 2. Mechanical ventilation not only ensures that a patient receives sufficient oxygenbut also helps to move carbon dioxide, a waste gas, out of the lungs. A person who cannot breathe efficiently on their own may retain carbon dioxide in the body, which can accumulate and reach toxic levels. It’s important to understand that ventilators are only used as life support; they do not treat or cure any medical conditions. In required time, the Artificial Intelligence technology take own decisions to provide proper ventilation for the respiratory patients from the ventilator. The output is given to atmega8 microcontroller. In controller the analog values of ventilator is converted to digital values for further processing and this value is displayed in LCD display. The digitized output is sent to transceiver for wireless transmission. The received values are given to PC via RS232 interconnection.

4.2 HARDWARE COMPONENTS

Among the basic components of an AI ventilator, there are ambu bag,MG995 servo motor,LM35DT sensor, driver, MAX30100 sensor, SPO2 sensor, flow sensor, LCD display, Timer, Microcontroller and power source.

4.2.1 Ambu bag



AMBU bag, also known as a manual resuscitator or bag valve mask (BVM), is **a device used to provide respiratory support to patients in emergency and non-emergency situations**. It consists of a self-inflating bag, a mask or mouthpiece, and a valve to control the flow of air.

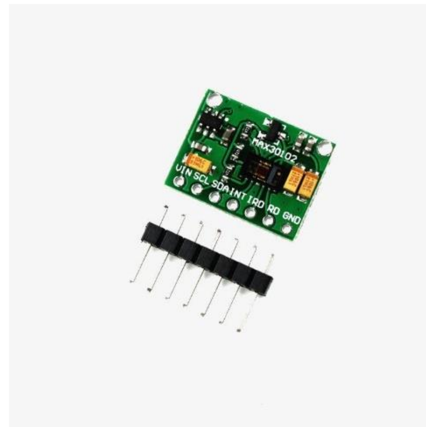
.

4.2.2 MG995 SERVO MOTOR

The G995 Metal Gear Servo Motor is **a high-speed standard servo can rotate approximately 180 degrees (60 in each direction) used for airplane, helicopter, RC-cars and many RC model**. Provides 10kg/cm at 4.8V, and 12kgcm at 6V. It is a Digital Servo Motor which receives and processes PWM signal faster and better.

Fig 4.2 MG995SERVO Motor

4.2.3. LM35DT sensor and MAX30100 sensors



LM35DT Atemperature sensor is **a device that detects and measures hotness and coolness and converts it into an electrical signal**. SpO2 sensors measure your blood oxygen saturation or put more simply, the amount of oxygen you have in your blood. In LM35DT measures temperature. MAX30100 is an integrated pulse oximetry and heart- rate monitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to **detect pulse oximetry and heart-rate signals**.

4.2.4 (20*4) LCD Display with I2c convertor

The LCD's are lightweight with only a few millimeters thickness. Since the LCD's consume less power, they are compatible with low power electronic circuits, and can be powered for long durations. Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. When the LCD is in the off state, light rays are rotated by the two polarizer and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent. When sufficient voltage is applied to the sensors, the liquid crystal molecules would be aligned in a specific direction.

Fig 4.4 LCD display   Table 4.5 Display pins

When the LCD is in the off state, light rays are rotated by the two polarizer and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent. When sufficient voltage is applied to the sensors, the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarizer, which would result in activating/highlighting the desired characters. In our project LCD display is used to display the digital value Spo2 values.

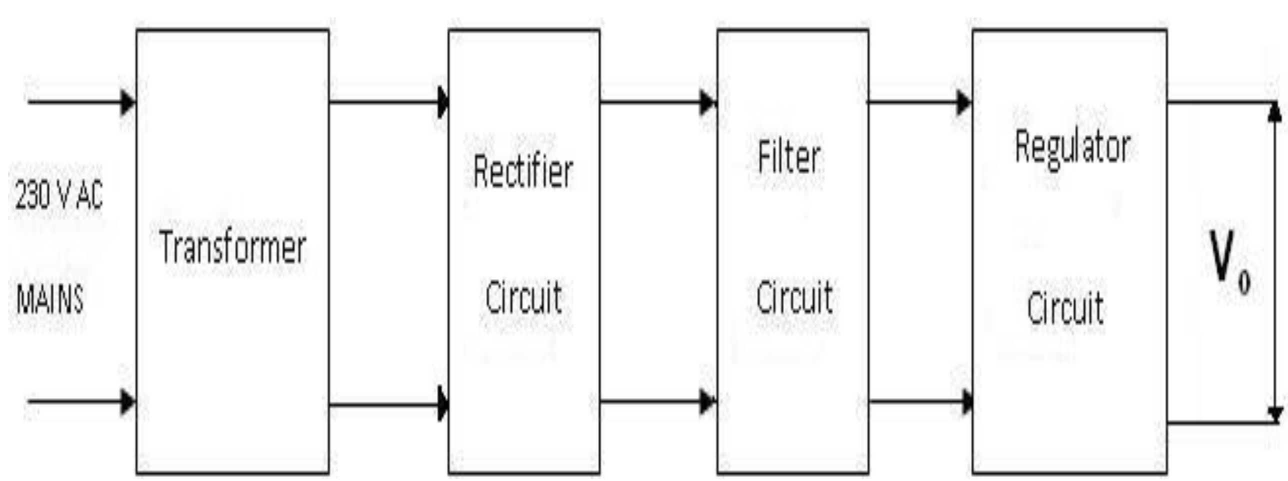
4.3 POWER SOURCE

4.3.4 12v 2amp and 11v/1 amp adapter and 10k port

This 12V and 11 V DC power supply can **provide a power source for any electronic device that has an M-type barrel and requires 12 and 11 volts and up to 1 to 2 amps of power**. The power supply is a perfect replacement for the lost or damaged 12-volt 2-amp power adapter you've used to power your existing electronics

A 10k potentiometer (a.k.a "pot" or "knob") is an electronic component that can be used to **control the flow of electricity through a circuit**, much like a faucet regulates the flow of water in your home.

Fig 4.7 Block diagram of power supply



The given block diagram includes following:

Rectifier

The function of the rectifier is to convert AC to DC current or voltage. Usually in the rectifier circuit full wave bridge rectifier is used. It flows only one direction. The reverse operation is performed by an inverter.

Filter

The Filter is used to remove the pulsated AC. A filter circuit uses capacitor and inductor. The function of the capacitor is to block the DC voltage and bypass the AC voltage. The function of the inductor is to block the AC voltage and bypass the DC voltage.

Voltage Regulator

Voltage regulator constitutes an indispensable part of the power supply section of any electronic systems. The main advantage of the regulator ICs is that it regulates or maintains the output constant, in spite of the variation in the input supply.

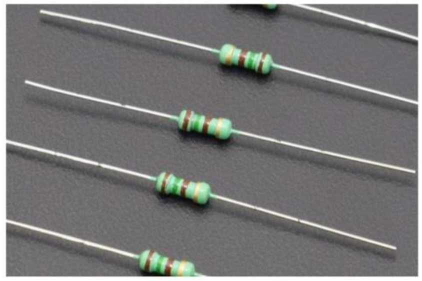
Voltage Regulation

- Two basic categories of voltage regulation are:
  - ❑ line regulation
  - ❑ load regulation

4.3.4.1 The purpose of line regulation is to maintain a nearly constant output voltagewhen the input voltage varies.

4.3.4.2 The purpose of load regulation is to maintain a nearly constant output voltage when the load varies

4.4 CAPACITOR,RESISTOR,THERMISTOR:



A **resistor** is **an electrical component that limits or regulates the flow of electrical current in an electronic circuit**

A ***thermistor*** is a semiconductor that contains greater resistance material than conducting material and a resistor that reacts to temperature.

25V 1000uF **Electrolytic *Capacitor* from *Keltron*.** Its suitable for **power supply filter, home theater systems, power supplies, audio amplifiers, and hobby project applications up to 25**

4.5 MICRO CONTROLLER
ARDUINO UNO

The microcontroller selected for this project is an Arduino UNO. It is one of the larger Arduino boards with 12 PWM (Pulse Width Modulation) channels, 54 digital I/O (Input/Output) pins, 12 analog inputs and two analog outputs. This board is powered by an ARM (Advanced RISC (Reduced Instruction Set Computing) Machines) processor. The smaller size of the ARM processor, its reduced complexity, and lower power consumption makes the Arduino Due suitable for miniaturized devices such as the AI VENTILATOR. Arduino Due is most suitable for this project due to its powerful ARM core and large Random Access Memory (RAM) (96 kb) and Read Only Memory (ROM) (512 kb). This project requires a number of General-Purpose Input/output (GPIO). In case of this project the number of input and output are from the motor, the motor driver, the LCD display, the pressure sensor, the optical sensor and the buzzer, hence the Due is most suitable choice. In Arduino at mega 8 microcontroller was used in this project.

4.6 NODE MCU

Parts of Node MCU Development Board

ESP 12-E Module

The development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS.

There's also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IOT devices nowadays.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a Wi-Fi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 Node MCU even more versatile.

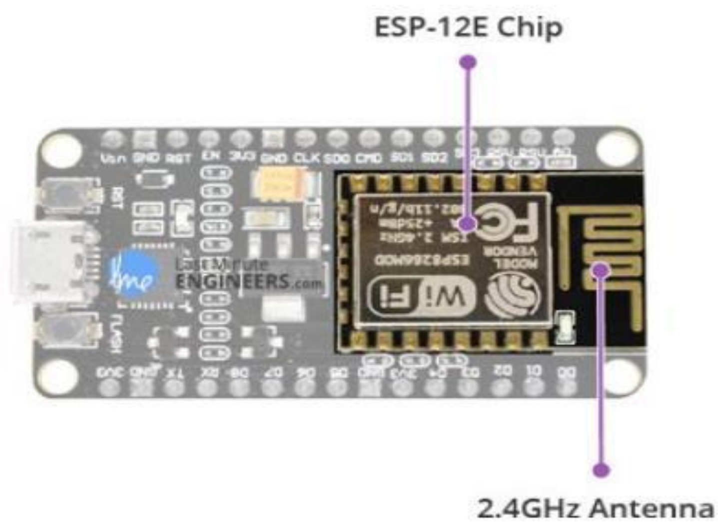
• Tensilica Xtensa

® 32-bit LX106

 80 to 160 MHz clock frequency

 128 kb internal RAM

 4 MB external flash



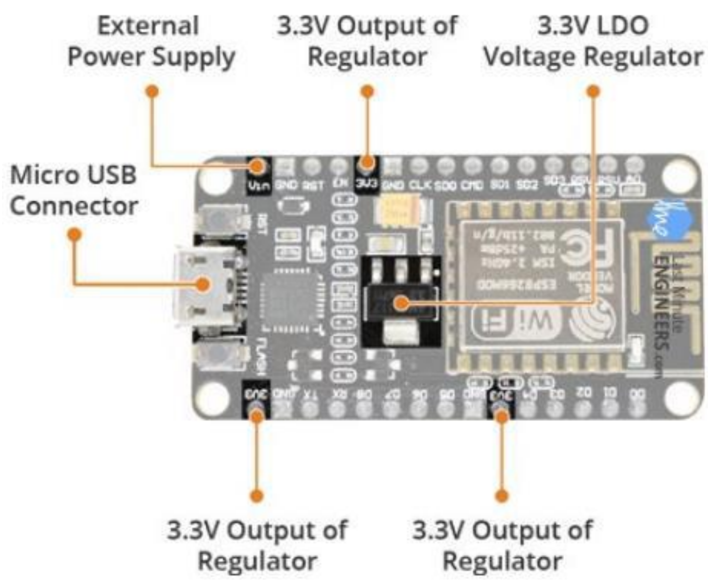
 802.11b/g/n HT40 Wi-Fi transceiver

Figure 2. ESP 12E module in Node MCU Development board

POWER SUPPLY

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labelled as 3V3. This pin can be used to supply power to external components.

Power to the ESP8266 Node MCU is supplied via the on-board Micro B USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin can be used to directly supply the ESP8266 and its peripherals.



Power module on a Node MCU development board

- Operating voltage 2.5V to 3.6V
- On-board 3.6V 600mA regulator
- 80 mA operating current
- 20 µA during sleep mode

Peripheral I/O

The ESP8266 Node MCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

 ADC channel – A 10-bit ADC channel.

 UART interface – UART interface is used to load code serially.

 PWM outputs – PWM pins for dimming LEDs or controlling motors.

 SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors and peripherals.

- I2S interface – I2S interface if you want to add sound to your project.

As a result of the pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin), a single GPIO pin can act as PWM/UART/SPI.

BUZZER

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren

CHAPTER 5

SOFTWARE INSTALLATION

5.1 INTRODUCTION

To design a mechanical ventilator with reduced cost possible bysimplifying the design for the mechanical ventilation process. Make a simulation model using MATLAB/Simulink software to perform experiments, predict results, and provide a similar interface. The automatic and simplify the process to replace the need for skilled professionals/therapists for operating patient's conditions on regular basis by providing it with decision-making ability. The CREATIS team has recently developed a semi-automatic segmentation software, allowing the calculation of the lung volume subjected to cyclic hyperinflation within a few minutes. This semi-automatic segmentation can be performed by a non-expert medical operator. Researchers have so far analyzed the results of 20 ARDS patients with promising preliminary results.

5.1.1 Installation of Node MCU

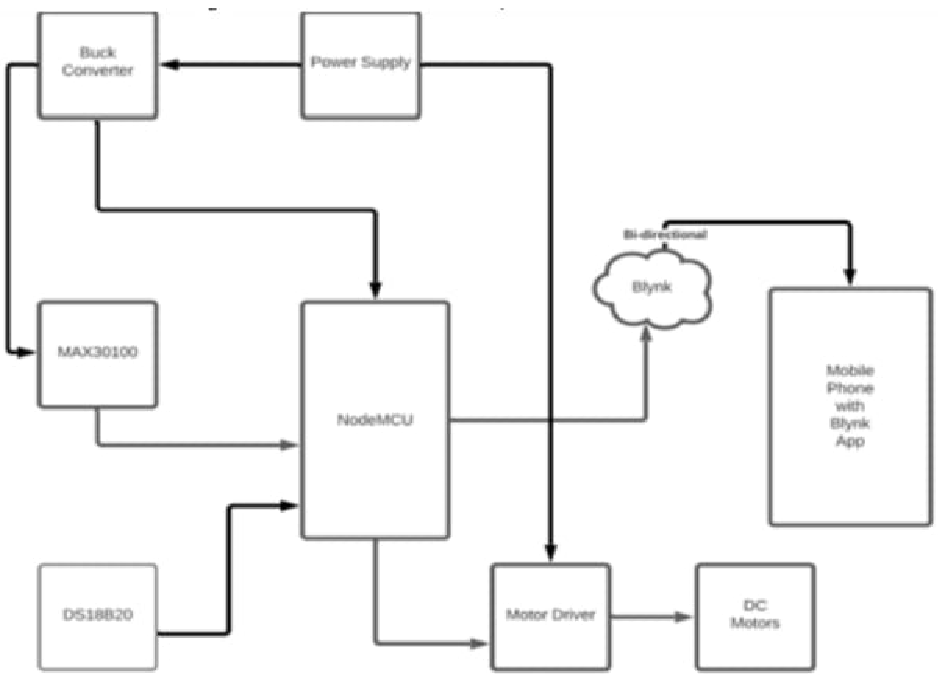
Node MCU is basically, an *Open Source* **IOT** (i.e. Internet of Things) device, which is similar to *Arduino*, and can be **coded** using an **Arduino IDE**, but it's not Arduino, not even close to it, as **Arduino** is all about **Hardware** power, and **Node MCU** is more like **Software power**, Arduino has more IO ports and Better Compatibility all over, on the other hand, Node MCU has *better RAM* and *Storage*, and it has inbuilt **WIFI** module. So that's a little Overview of our Board.

We will display the SpO2 and BPM value in 0.96″ OLED Display. With each beat, the display value is changed in the OLED screen. By using Bluetooth module HC-05/HC-06 (operating in a slave mode), we can send data to the android app wirelessly and monitor the data on the app as well as keep a track record of the data in text format. In this way, we can send the data read from the device to another device or to the Internet. This wearable device can be used by athletes to monitor their heart rate and blood oxygen levels during a workout.

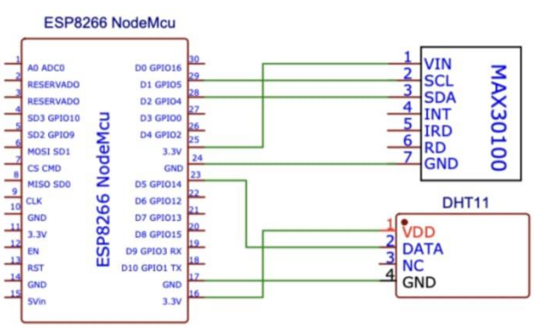
5.2 BLOCK DIAGRAM

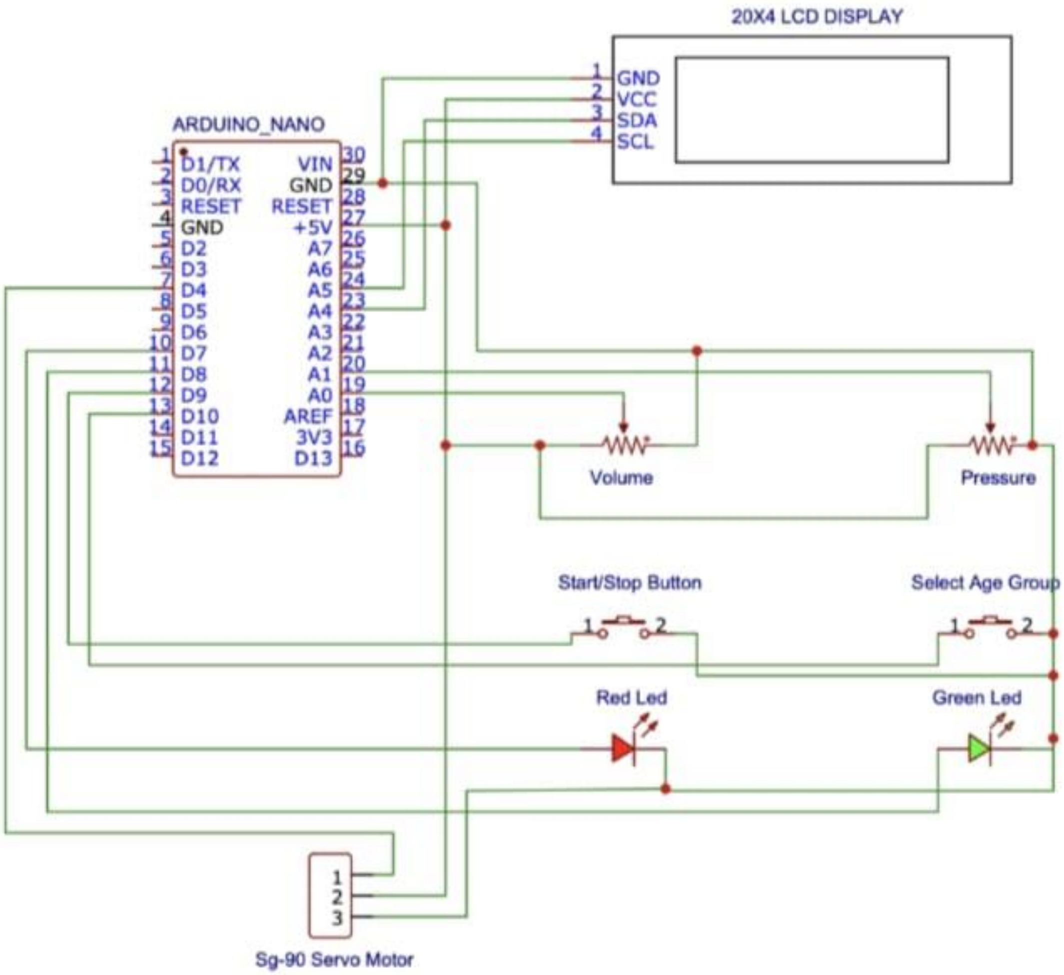
The block diagram gives the functionality of the overall project. The Node MCU unit is the microcontroller or the main controlling unit of the system. The user uses the mobile application in setting commands for functioning of the appliances.

The mobile application interprets the command form in user in voice or switch mode and sends signal to the Node MCU unit, over a wireless network established by Wi-Fi communication. Hence the Wi-Fi module (actually inbuilt into Node MCU), helps the microcontroller establish Wi-Fi communication with a device and take commands from an application over wireless network. The Node MCU on further receiving the signal then turns on/off the appliance with the help of relay. The Node MCU, relay and the final appliances are physically connected. There is a power supply unit that powers the microcontroller, the relay as well as the final appliances. There is also a display unit that displays the status of the application



5.2.1 CIRCUIT DIAGRAM





5.3 HARDWARE MODELLING AND SETUP

Driver installation for hardware interfacing

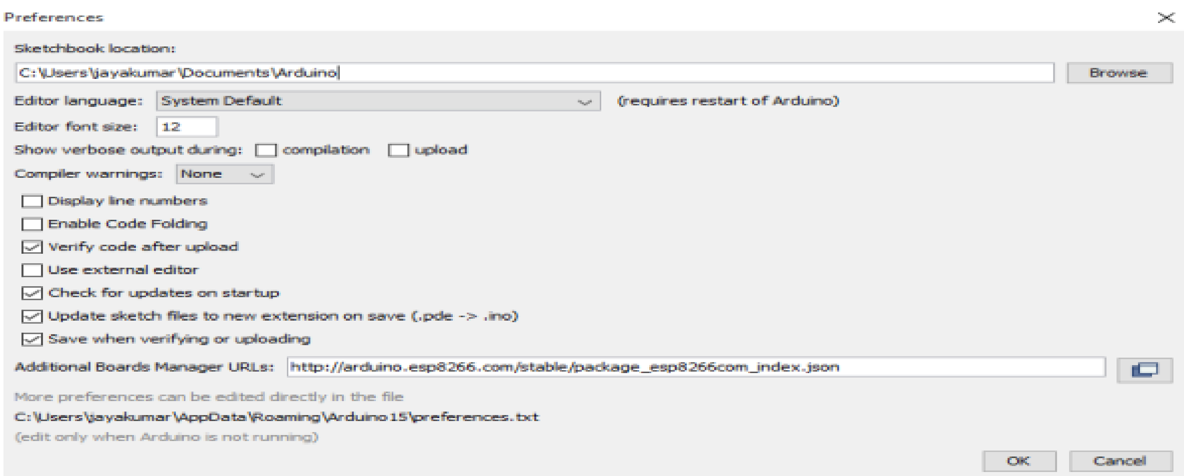
Mostly these days devices download and install drivers on their own, automatically. Windows doesn't know how to talk to the USB driver on the Node MCU so it can't figure out that the board is a Node MCU and proceed normally.

 Node MCU Amica is an ESP8266 Wi-Fi module based development board. It has got Micro USB slot that can directly be connected to the computer or other USB host devices. Ti has got 15X2 header pins and a Micro USB slot, the headers can be mounted on a breadboard and Micro USB slot is to establish connection to USB host device. It has CP2120 USB to serial converter.

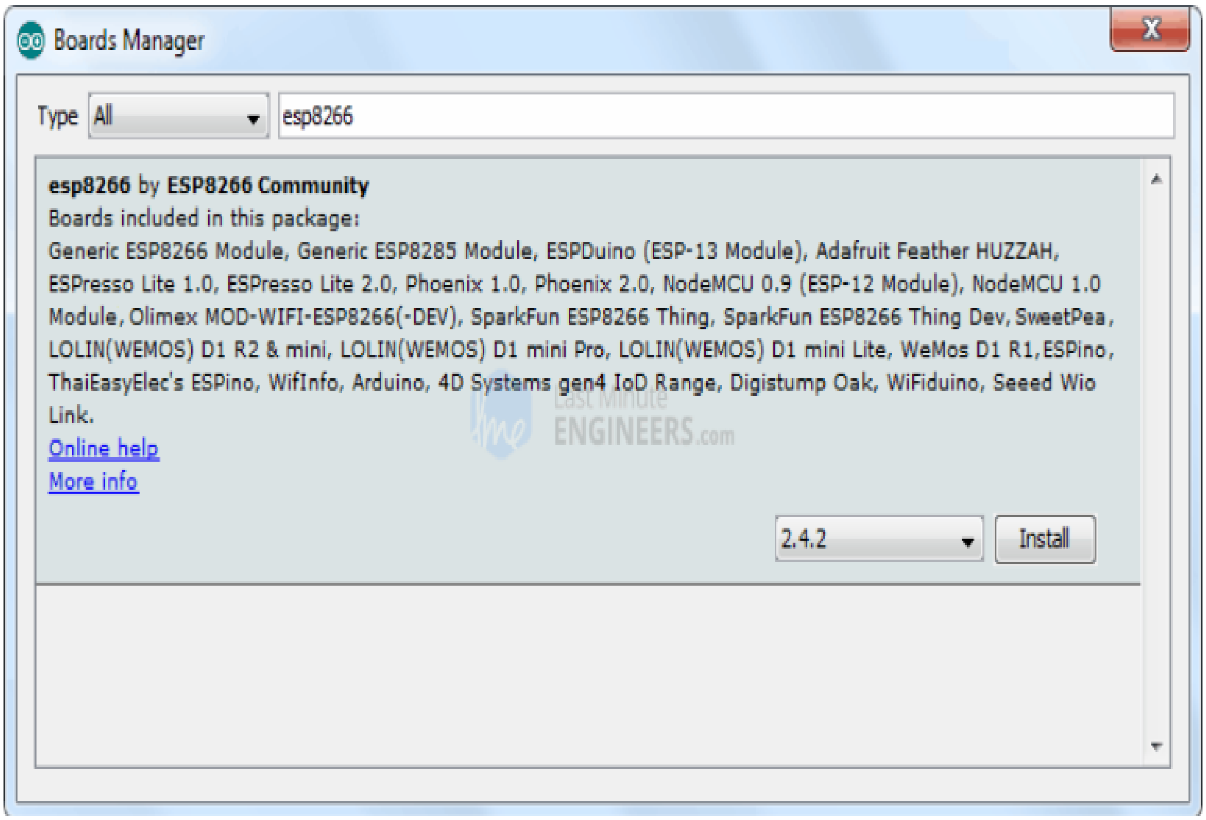
- In order to install CP2120 (USB to serial converter), user is needed to download the driver for the same.
- Once user downloads drivers as per its respective operating system, the system establishes connection to Node MCU.
- The user needs to node down the COM post allotted to newly connected USB device (Node MCU) from device manager of the system. This com port number will be required while using Node MCU Amica.

5.3.1 Interfacing Node MCU with Arduino IDE

To begin with the latest Arduino IDE version, we’ll need to update the board manager with a custom URL. Open up Arduino IDE and go to File > Preferences. Then, copy below URL into the Additional Board Manager URLs text box situated on the bottom of the window:
http://arduino.esp8266.com/stable/package_esp8266com_index.json

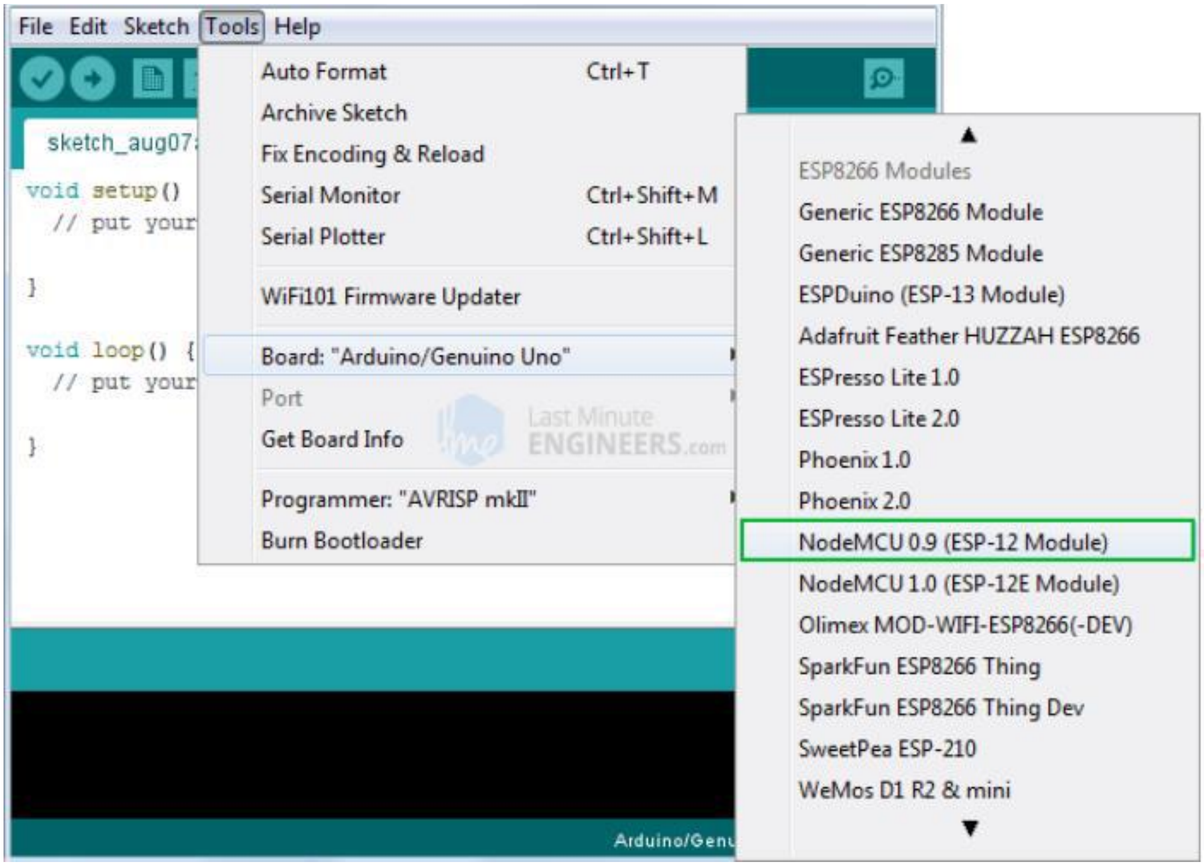


OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Filter your search by typing esp8266. Click on that entry and select Install.



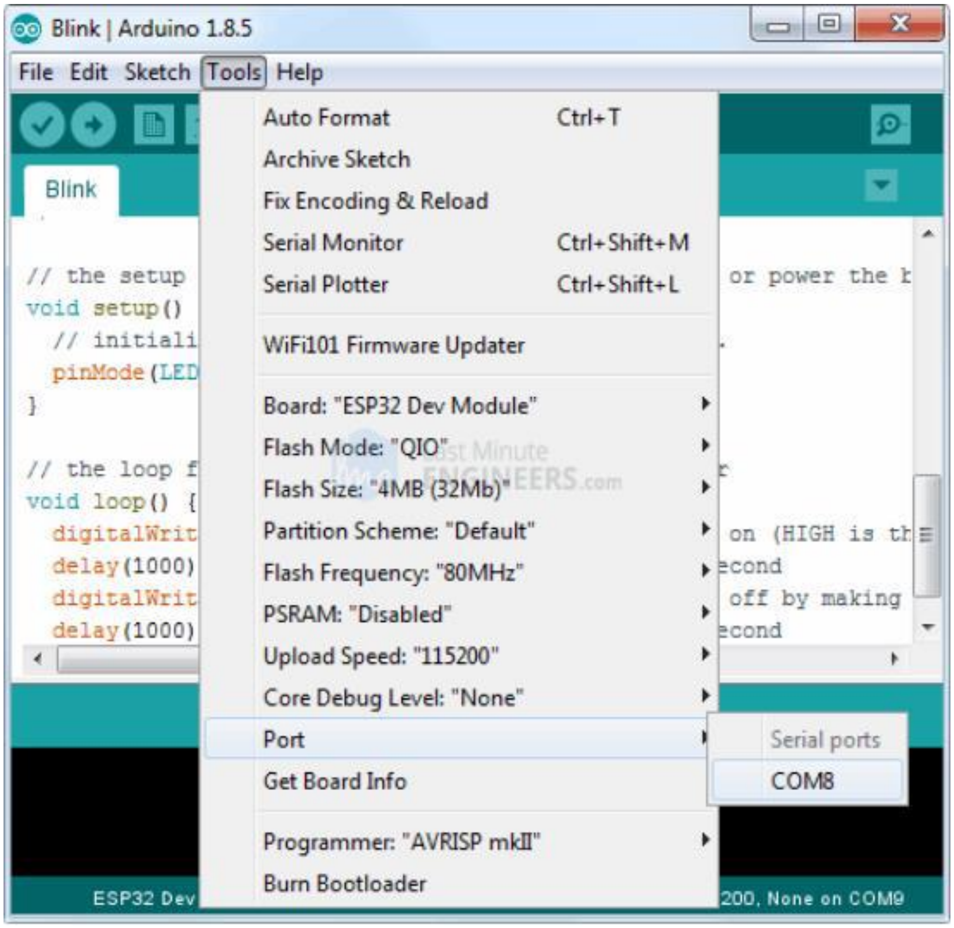
. ESP8266 board installation in Arduino IDE.

Before we get to uploading sketch & playing with LED, we need to make sure that the board is selected properly in Arduino IDE. Open Arduino IDE and select Node MCU 0.9 (ESP-12 Module) option under your Arduino IDE > Tools > Board menu.



Arduino IDE board manager installation

Now, plug your ESP8266 NodeMCU into your computer via micro-B USB cable. Once the board is plugged in, it should be assigned a unique COM port. On Windows machines, this will be something like COM#, and on Mac/Linux computers it will come in the form of /dev/tty.usbserial-XXXXXX. Select this serial port under the Arduino IDE > Tools > Port menu. Also select the Upload Speed: 115200



Assigning communication port on Arduino IDE.

5.3.2 Uploading code to Node MCU

- NodeMCU is connected to PC using a USB cable.

- Now, we'll set up the Arduino IDE by changing some settings. So, open up the Arduino IDE. Select Tools > Board and select 'NodeMCU 1.0 (ESP-12E Module)' as the board. And that's all the settings we need to change. So now we begin writing the code.

- Select Files > Examples > Blynk > Boards_WIFI > ESP8266_Standalone. A new file with some prewritten code opens. The following changes to the code are made.

1. The line which says 'char auth[] = "YourAuthToken", replace YourAuthToken part with your Blynk's authentication token that was generated by the Blynk server.

2. The line which says *char ssid[] = "YourNetworkName"*, replace *YourNetworkName* part with the name of Wi-Fi network that the Node MCU must connect to.

3. The line where it says *char pass[] = "YourPassword"* and replace the *YourPassword* part with the password of the Wi-Fi network.

Figure 21. Code in Arduino IDE to be installed to Node MCU

• The code is ready to be uploaded to the hardware. On clicking upload button, the code is uploaded to Node MCU and the next time it's powered on, it automatically connects to the assigned Wi-Fi network.

5.3.3 Installation and setup of IFTTT

- To configure IFTTT we visit their website https://ifttt.com and sign up using google account.
- After signing in, we select *My Applets* from header, and select *New*. Search for *Google assistant* and connect. Allow IFTTT for permission to use Google account to add voice commands to it.
- Configure the application to work as desired, and Create Trigger

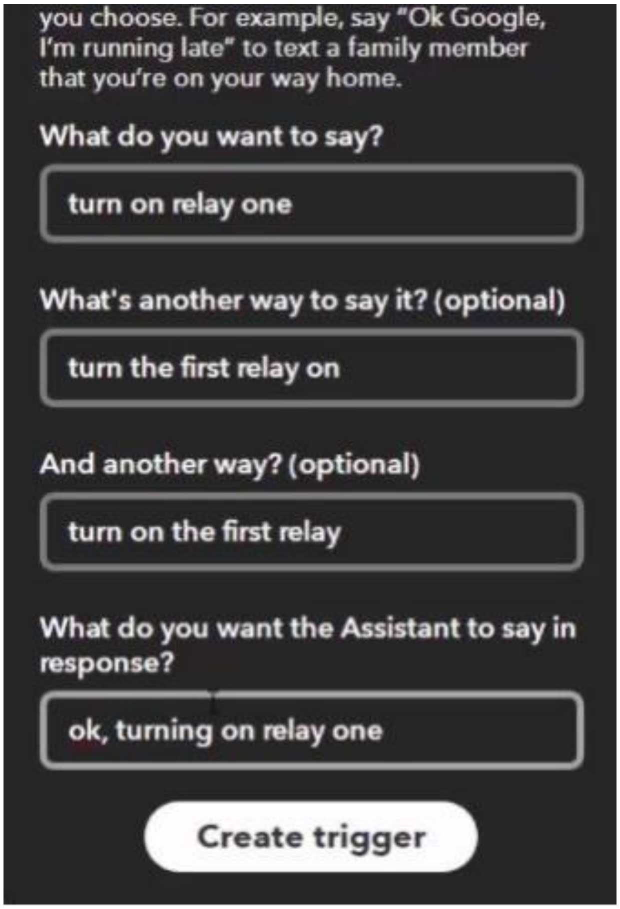


Figure 22. IFTTT configured with actions and commands.

• Select webhooks that will allow to send commands to Blynk server.
Add
http://188.166.206.43/ YourAuthTokenHere / update / DigitalPinToBeUpdateHere to the URL field.

YourAuthTokenHere is replaced by the authentication token generated by Blynk server. *DigitalPinToBoUpdatedHere* is replaced by the digital pin of Arduino that corresponds to the Node MCU rather than the one of Node MCU itself.

Following details are added to program the applet. Here '0' means to turn on, so we are basically saying Blynk to turn on relay that is connected to pin D3, which in our case is relay one.
Click on *Create Action* and finish.

Figure 23. Configuration of applet to switch relay with voice commands

- Similarly, another applet is created to turn off the relay, repeating all the steps above except the following changes: instead of writing “Turn on relay one”, written “Turn off relay one”and instead of [“0”], written [“1”]. Two triggers are created to turn on and off one Relay.

- Similarly, we create triggers for remaining 3 relays by change the phrase and Digital pin for each Relay. All the other steps will remain the same.

In the end for 4 relays, we have 8 triggers to turn each of them on or off. After all this is done, voice commands to Google Assistant can switch relay.

5.4HARDWARE ASSEBLY

Hardware assembly mainly includes connecting specific digital pins of NodeMCU to the 4 relays on the relay module, including the connection of supply and ground pins. The main functional assemble in this prototype is simple. The further 4 relays are fit to be connected to any appliance desired to be controlled.

The vital part in hardware assembly is taking into account the digital pin that corresponds to which relay. This connection is done as per the setup of Blynk application. The radio buttons on Blynk application are set up to switch a particular digital pin in Node MCU. It is made sure that the relay connection are physically made according to this set up. For example, we have assigned the radio button on Blynk application corresponding to relay 1 to work with D3. Then physical connection of relay 1 is made with D3 of Node MCU.

In this prototype instead of real home appliances, we connect the relays to LEDs, (according to circuit diagram) to just ensure the functionality of the prototype. The prototype is given a supply from a 9V battery.

5.5 LOGIC AND OPERATION FLOW CHART

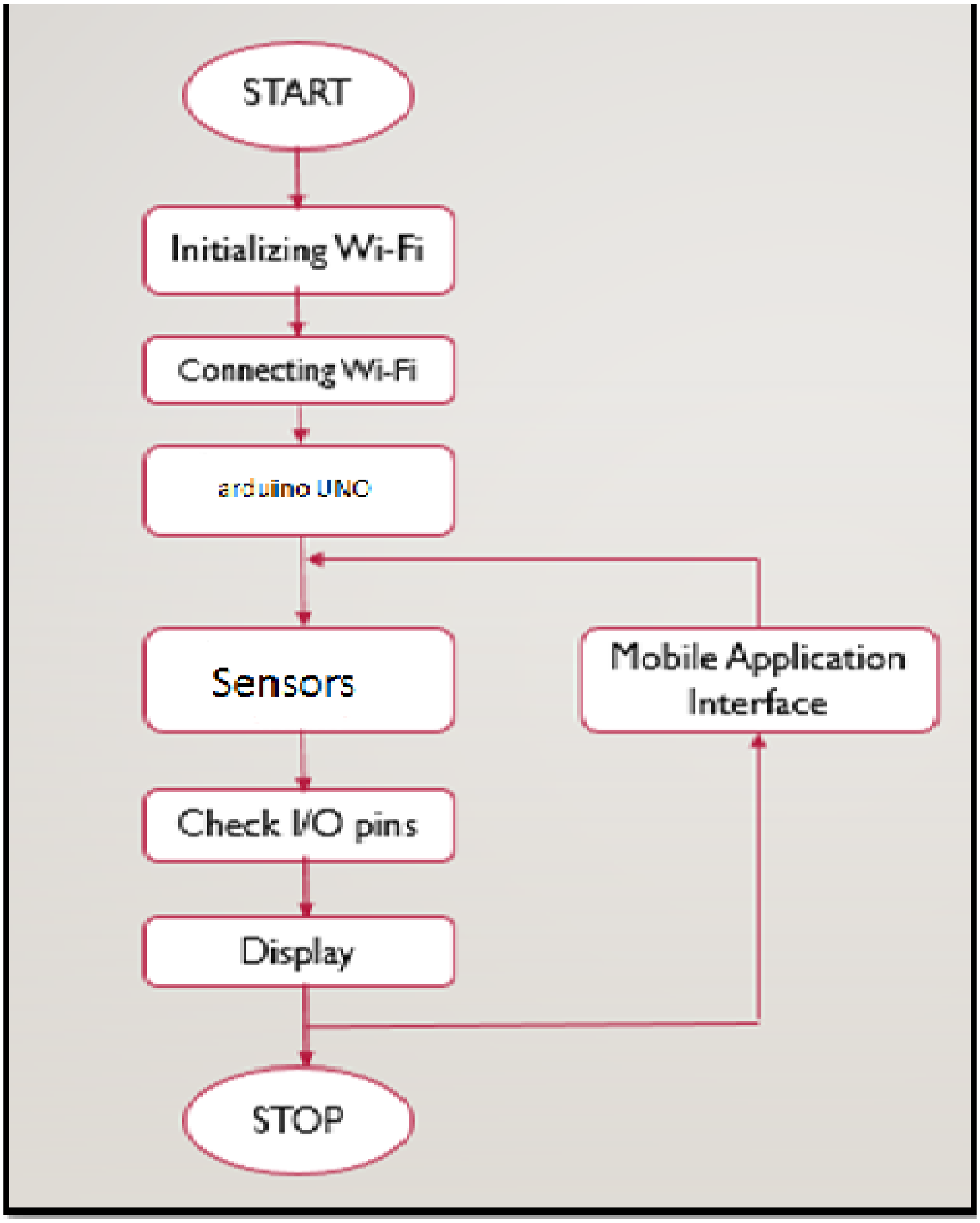


Figure 25. Flow chart of prototype function.

This flow chart shows the working of the project. The process starts be initializing the Wi-Fi, the network name and password are written in the code and uploaded to Node MCU. The android device is connected to Node MCU over Wi-Fi. The Blynk server is set up and connection is made, the devices is identified in the Blynk server using the generated authentication token. The command for controlling the load is given to the application, and this command, over Wi-Fi network is sent to the Node MCU.

5.6 PRINCIPLE AND OPERATION

Node MCU is an open source IOT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term “Node MCU” by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

5.6.1 Advantages of Node MCU

- Low cost, the Node MCU is less costly compared to any other IOT based device.
- Node MCU has Arduino Like hardware I/O. It is becoming very popular in these days that Arduino IDE has extended their software to work in the field of ESP 8266 Field module version.
- Node MCU has easily configurable network API.
- Integrated support for Wi-Fi network: ESP 8266 is incorporated in Node MCU, which is an easily accessible Wi-Fi module.
- Reduced size of board.
- Low power consumption.

5.6.2 Disadvantages of Node MCU

- The operation of the circuit depends on the working internet connection. If the working internet connection is not available then it will not run.
- Node MCU also depends on the free server provided by the third party, if the free server is not working then it will not run.
- Node MCU has less resources of official documentation
- Need to learn a new language and IDE
- Reduced pinout
- Scarce documentation

5.7 WIRELSS COMMUNICATION NETWORK

The prototype aims to wireless control over home appliances with the technology of IOT. As discussed earlier, IOT supports various wireless communication protocols, like Bluetooth, Z-Wave, Zigbee etc. this prototype uses Wi-Fi as wireless communication network to establish remote access over home appliances. This is because Wi-Fi has its own advantages over other wireless communication protocols.

CHAPTER 6

SOFTWARE AND METHODOLOGY

INTRODUCTION

To design a mechanical ventilator with reduced cost possible bysimplifying the design for the mechanical ventilation process. Make a simulation model using MATLAB/Simulink software to perform experiments, predict results, and provide a similar interface. The automatic and simplify the process to replace the need for skilled professionals/therapists for operating patient's conditions on regular basis by providing it with decision-making ability. The CREATIS team has recently developed a semi-automatic segmentation software, allowing the calculation of the lung volume subjected to cyclic hyperinflation within a few minutes. This semi-automatic segmentation can be performed by a non-expert medical operator. Researchers have so far analyzed the results of 20 ARDS patients with promising preliminary results.

The next step is to integrate this tool into a ventilator adjustment strategy based on quantitative imaging analyzed in real time. The new challenge for these researchers is the segmentation of the pulmonary parenchyma including non- aerated areas, whose tomographic density is close to that of the surrounding tissues(chest wall, mediastinum) on non-injected scans; the absence of contrast defeats algorithms based solely on scan density. The development of algorithms based on machine learning may be an interesting avenue, integrating the constraint of result availability within a few minutes;

The evolution of ARDS takes place over a maximum of one week and image-based protective ventilation strategies must be applied from the very first hours, in order to expect an effect on the prognosis. Simulation model used was made with MATLAB/Simulink software to provide a computational model to simulate the interaction between the mechanical ventilator and human lungs.

The model parallels the hardware and software design of the actual Mechanical Ventilator prototype. It includes all the parameters and variables with help of Simscape libraries which were used while simulating the model. This assists medical professionals in better visualization of the condition and thus helps in decision-making.

6.1CODE VISION AVR

Code Vision AVR is the only Integrated Development Environment on the market that features an Automatic Program Generator (Code Wizard AVR) for the AVR8, AVR8X, AVR DA, AVR DB, AVR DD and XMEGA chips. Code Vision gathers various metrics for types and type members and displays this information above or after their declarations. Position of metrics, their maximum number, and other preferences can be configured on the Editor | Code Vision page of Jet Brains Rider settings Ctrl+Alt+S.

6.1.1System Requirements of CODE VISION AVR

It is compatible with Windows® Vista, 7, 8, 10 and 11, 32 and 64-bit operating systems. For the Extension to be installed correctly, Microchip Studio 7 must be already present on the computer, before the CODEVISIONAVR installer is launched. The CODE VISION AVR is only Integrated Development Environment on the market that features an Automatic Program Generator (Code Wizard AVR) for the AVR8, AVR8X, AVR DA, AVR DB, AVR DD and

XMEGA chips. To add a new library to CODE VISION AVR, you should go to the Project/Configure/C Compiler/Paths and then pass in your library path. Install the Code Vision AVR C Compiler in the default directory: C:\cvavr. Install the Atmel AVR Studio debugger in the default directory: C:\Program Files\Atmel\AVRStudio. The demonstration program to be developed in the next few pages requires an Atmel AT90S8515 microcontroller and the STK500 starter kit.

6.1.2Micro controller of CODE VISION AVR

AVR studio is an Integrated Development Environment (IDE) developed by ATMEL for developing different embedded applications based on 8-bit AVR microcontroller. Before the installation of AVR Studio you have to install the compiler WinAVR. By installing WinAVR allow AVR Studio to detect the compiler.

6.1.3AUTOMATIC VOLTAGE REGULATOR (AVR)

AVR is an electronic device that maintains a constant voltage level to electrical equipment on the same load. An automatic voltage regulator (AVR) is a device used in generators with the purpose of automatically regulating voltage, which means that it will turn fluctuating voltage levels into constant voltage levels. The AVR regulates voltage variations to deliver constant, reliable power supply. We've explored 4 of the most common AVRs;

- servo
- magnetic induction
- static tap switching
- Ferro resonant

There are two types of voltage regulators: linear and switching. A linear regulator employs an active (BJT or MOSFET) pass device (series or shunt) controlled by a high gain differential amplifier. An automatic voltage regulator (AVR) is an electronic device that maintains a constant voltage level to electrical equipment on the same load.

6.1.4 AVR Working Principle

Automatic voltage regulators (AVRs) work by stabilizing the output voltage of generators at variable loads, but can also divide the reactive load between generators that are running in parallel (voltage droop), and helps the generator respond to overloads. The working principle of the AVR is to regulate the strengthening current in the exciter. If the generator output voltage is below the nominal voltage of the generator, the AVR will increase the amplification of the exciter.

Components of AVR

The major components of the AVR are the two closed-loop control systems and both of them have separate gate control, thyristor set, and DC excitation equipment. The limiter is used to control the excitation.

CODE VISION AVR Advantages

- There is no boot loader hence more flash memory as compared to arduino.
- Execution is fast since boot loader slows the startup time of a microcontroller.
- Extra GPIO pins, pin number 9 and 10 can be used as GPIO pins in AVR (provided, we are using internal RC oscillator).

CODE VISION AVR Applications

- Application that runs under Windows® Vista, Windows 7, Windows 8 and Windows 10, 32-bit and 64-bit
- Easy to use Integrated Development Environment and ANSI C compatible Compiler
- Editor with auto indentation, syntax highlighting for both C and AVR assembler, function parameters and structure/union members autocomplete.
- Supported data types: bit, bool, char, int, short, long, 64-bit long, float.
- Fast floating point library with hardware multiplier and enhanced core instructions support for all the new AT mega chips.

AVR Specific extensions

- Transparent, easy accessing of the EEPROM & FLASH memory areas, without the need of special functions like in other AVR compilers
- Bit level access to I/O registers
- Interrupt support
- Support for placing bit variables in the General Purpose I/O Registers (GPIOR) available in the new chips (ATtiny2313, ATmega48/88/168, ATmega165/169/325/3250/329/3290/645/6450/649/6490, ATmega1280/1281/2560/2561/640, ATmega406 and others)

6.2METHODOLOGY

The ventilator is connected to the patient through a tube (endotracheal or ET tube) inserted in the mouth or nose into the windpipe. When the doctor inserts the ET tube into the patient’s windpipe, the process is termed intubation. Some patients go through the surgical process in their neck and a tube is inserted via the hole created, this process is called tracheostomy.

The ventilator blows gas into a person's lungs according to the patient's requirement. It can take over all the respiration processor can assist the patient to breathe. The ventilator can provide a positive end-expiratory pressure (PEEP pressure) which helps to hold the lungs open, so the air sacs don't collapse. The ventilator should stop delivering oxygenated air as soon as the patient starts to breathe himself. If it operates while the respiratory system of the patient is active it may cause damage to the lungs and can cause serious injury to the lung (VILI). Thus, there must be a system to monitor when the patient starts to breath.

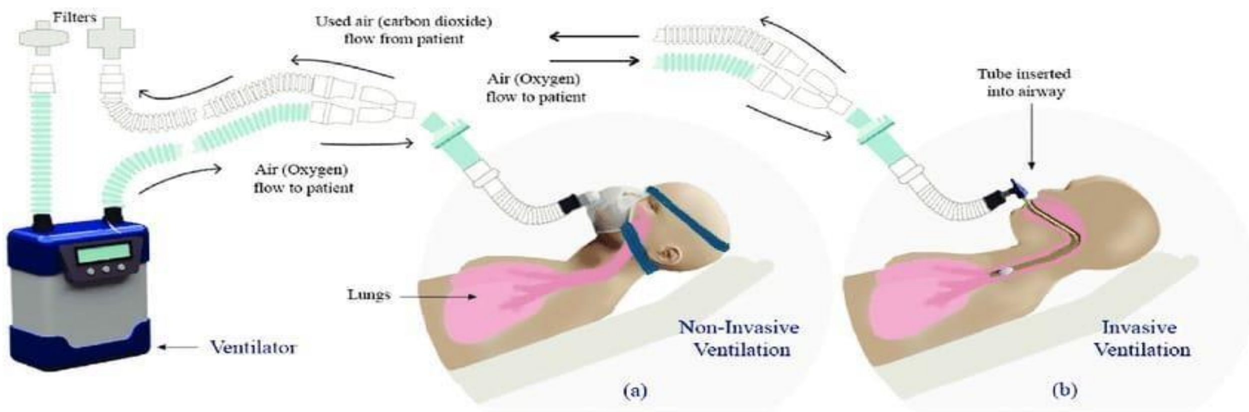


Fig 5.1 Methodology of Ventilator

6.2.1Open-Source Ventilator

An open-source ventilator is made by using freely-licensed design and is often thought to be used for disaster situation, the sources include the components that are freely available. Parts can be 3D-printed instead of purchasing to help keep the cost low. Documentation and testing of open-source ventilators has been going on since the start of the COVID-19 pandemic. Section 2.5.1 discusses the different types of low-cost ventilator systems that can be used to treat COVID-19 patients.

6.2.2Types of Open-Source Ventilators

There has been a lot of research going on designing low-cost open-source ventilators system since late 2020's. The conceptual designs of field portable ventilators for domestic and military emergency response exists, but there is limited information on how to reconstruct the ventilators as the software used for making them was not shared openly and some of the codes were written in assembly language.

1. Centralized Air Ventilation System

Centralized air ventilation systems are designed to supply air to several patients at once. In order to provide air to 10–20 patients at a time, an industrial oil-free air compressor is used. The air is mixed with oxygen, which is extracted from standard oxygen bottles. The air that is supposed to be received by each patient is regulated by individual valves. These valves are connected to microcontrollers and monitored for individual patients. This solution is suitable for intubated patients. The design of the terminal that allows individual patients to be connected can differ depending on the design.

2. Using a Blower for Ventilation Systems

This type of design uses a blower to feed the air to the individual patients. It is a compact and low-cost mask respirator. This solution has been designed and prototyped successfully. It is a low power device and integrated sensor for airway pressure can detect leakage and occlusion. This solution is suitable for patients that need to be moved from one place to another. It proposes such solution with a miniature turbine where oxygen and air intake aremixed in a single step. The housing box is 3D printed.

3. Artificial Manual Breathing Unit (AMBU) Bag Ventilation System Design

This type of design uses a standard AMBU bag to assist the patients with breathing that do not require intubation. This design is not based onconstant blower use. The AMBU bag can be automated to be able to adjust breathing rate and volume of air. It can also be made to regulate inspiration to expiration ratio and PEEP rate. Other design solutions include using a DC motor and a microcontroller by compressing a bag-valve mask (BVM) to achieve a higher degree of control over the respiration process. The Air Pump Type

This solution is similar to those design that uses a blower. It needs air pumps that are easily available like aquarium air pumps to supply necessary airflow. Meeting the maximum airflow requirements is one of the critical factors that these solutions need to ensure. Hence, it might not be suitable for the patients in dire conditions.

5.3VENTILATOR CONTROLLER LOOP

The next ventilator graphics that are important to understand are the loops. These graphics are one of the two variables, either pressure or flow, plotted against the volume during a breath. Each loop consists of an inspiratory and expiratory curve and allows for evaluation of respiratory mechanics. This is the Pressure/Volume loop.

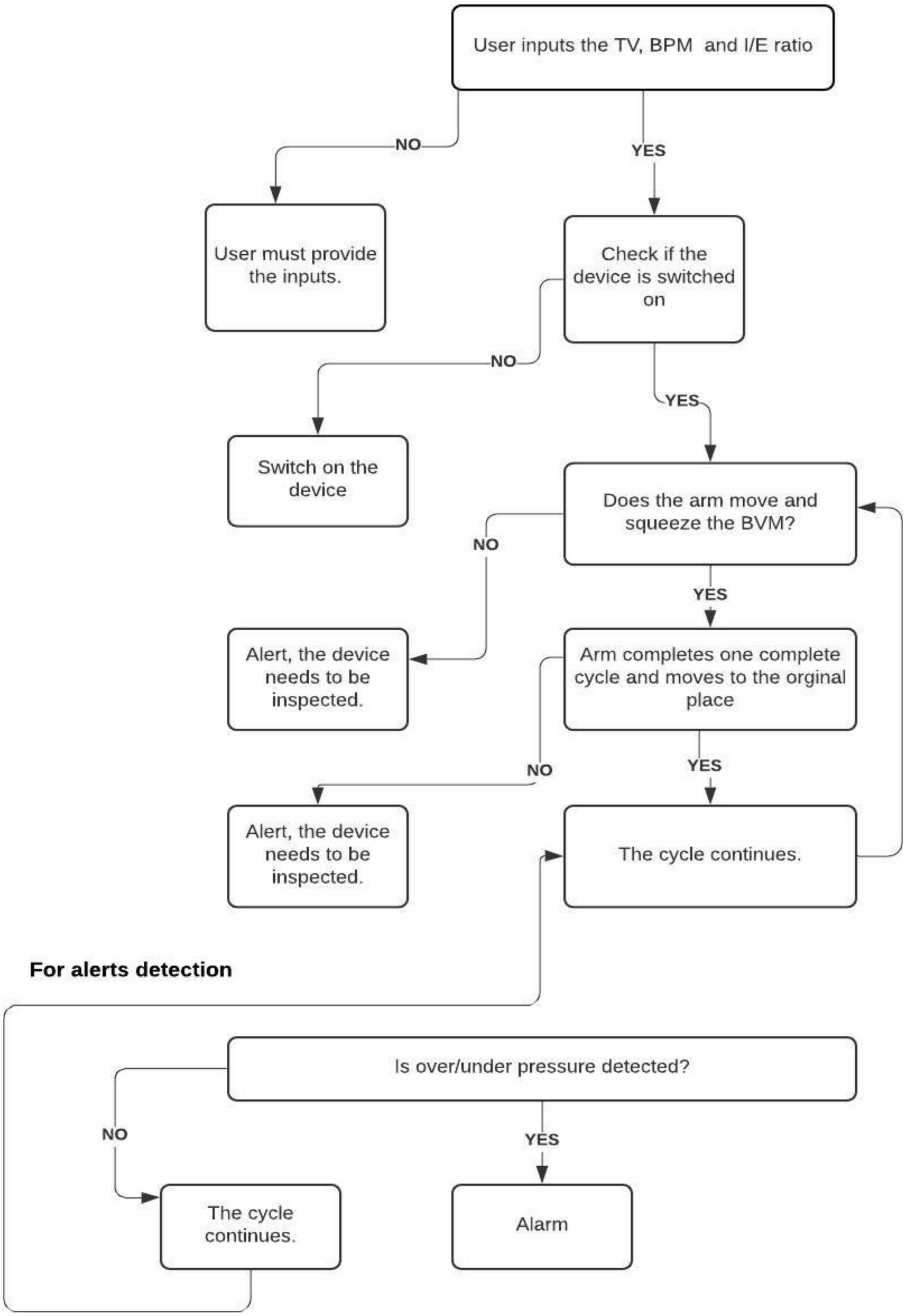


Fig 5.2 Ventilator Controller Loop

CHAPTER 7

7.1 RESULT

This chapter describes the tests that were conducted in the device after the flow sensor and alert codes were added to the microcontroller. The flow data were collected when the flow sensor was attached to the patient port of the BVM.It will monitor our heart rate,spO2 level ,temperature . The flow sensor reads real-time flow data. These flow data are important because they determines the airflow at the patient port of the BVM. The flow sensor was connected to the arduino, which monitored the ventilator performance parameters such as pressure, load, current, voltage and airflow. Flow data were collected every 30ms. The BVM and a Quick Lung breathing simulator (test lung) were connected.

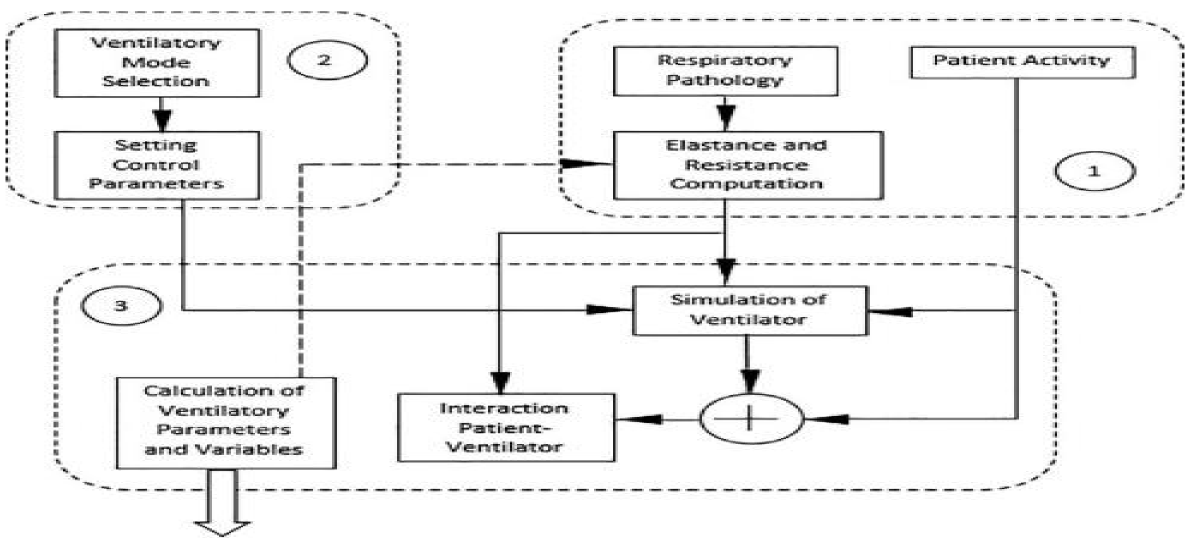
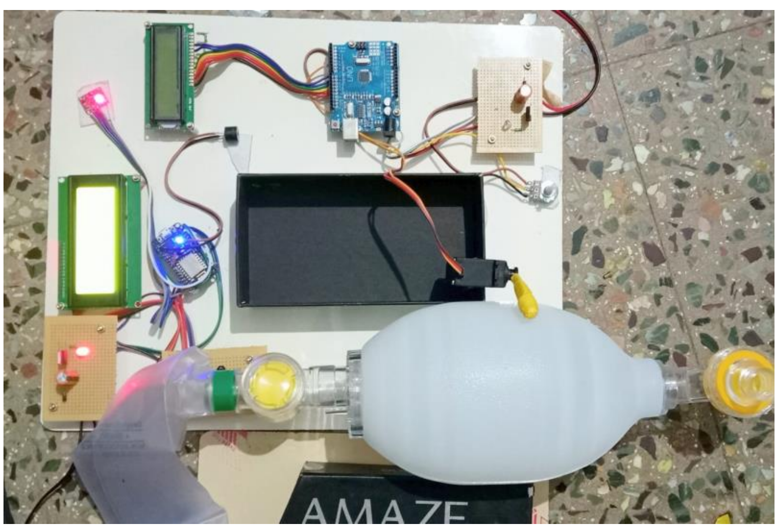


Fig 6.1 Block Diagram of Patient – Ventilator Interaction

The historical use of negative pressure Ventilator devices to modern positive pressure mechanical Ventilators, it has been a very crucial part of medical science. Advancements in biomedical engineering have developed the functionalities of ventilators. As a result, it has been an expensive treatment.

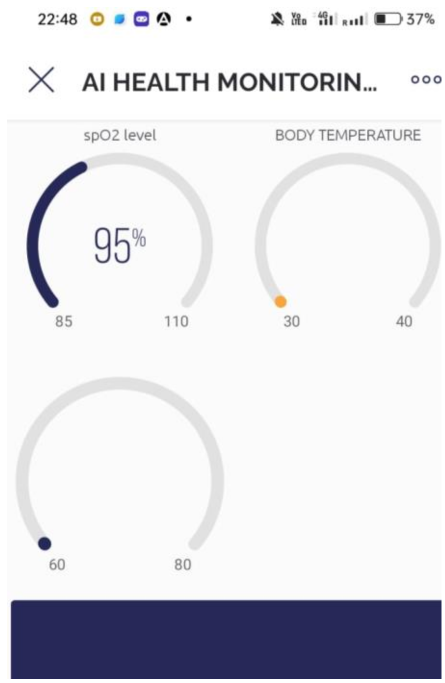
Due to the rapid increase in demand for mechanical ventilators in the COVID-19 pandemic, people with economically poor backgrounds find it difficult to afford the treatment. Also, the fees of medical professionals or therapists add monetary value. This research and reduce manual support approach would help in making the Artificial Intelligence Ventilator available for people. The simulation model of the AI ventilator is created to mimic the working of automatic ventilation with helpof AI Software. The results are quite similar to that of actual Ventilators. Simulation provides the freedom to study or experiment on a range of parametersor variables without any human life in danger. Also, the AI technology provide the model decision-making ability which will either assist the medical professionals in decision making or completely replace the need for human presence for monitoringpatients.

The concept is still under development. We will consider changes induced by the results of the prototype testing. Many cost-effective sensors that interact with the human body and ensure the accurate administration of respiration will also be added. Microcontrollers like arduino will be used along with the simulated model for smooth conduction of the ventilation process.



AI HEALTH MONITORING AMBU VENTILATOR MODEL

7.2OUTPUT



IF TEMPERATURE ,SPO2 ,HEART RATE goes beyond above normal level buzzer Sound will automatically indicated

7.3 COST ESTIMATION

COMPONENTS	AMOUNT
AMBU BAG	2100
Arduino UNO	400
ESP8266 wifi module	330
LCD 20*4 display	370
I2c converter module	100
LCD 16*4 display	300
LM35DT Temperature sensor	100
MAX30100	170
MG995 servo motor	315
Zero pcb board	60
10k port	30
25v 1000uf keltron capacitor	40
Resistor	40
Thermistor	30
Female cable	50
12v/2amp adapter	160
11v/1amp adapter	120
Buzzer	50
Ribbon cable wire	20
Plain board	180

TOTAL = 4962

CHAPTER 8

CONCLUSION and FUTURE ENHANCEMENT

CONCLUSION

Our project intends to Artificial Intelligence (AI) has the potential to personalize mechanical ventilation strategies for patients with respiratory failure and monitor heart rate ,spO2 level,temperature. However, current methodological deficiencies good limit clinical impact. We identify common limitations and propose potential solutions to facilitate translation of AI tomechanical ventilation of patients and various sensors that were considered for the design were discussed and added a new technology of AI. To build low-cost ventilator device such as Artificial Intelligence based ventilator, which can be made with readily available components and can provide a reliable ventilator solution, can be used in case of surge demand. The relative low cost of this project could potentiallyprovide a ventilator solution in other resource constricted environments as well.

8.1FUTURE ENHANCEMENTS

- To assist with mini ventilation to remove CO2.
- Multiple modes of ventilation such as pressure control, assist control, can be included to make the system more intelligence.
- To provide FiO2 and maintain End-expiratory lung volume with positive End-expiratory pressure (PEEP), to maintain oxygenation.
- User factors can be improved by adding a touch screen display device to enhance the system.

APPENDICES

```
#include <ESP8266WiFi.h>;#include <WiFiClient.h>; #include <ThingSpeak.h>; int
WiFiClient client;

const char* ssid = "Active Galaxy"; //Your Network SSID

const char* password = "Smilepls@8" ; //Your Network Password

unsigned long myChannelNumber = 982122;//YourChannel Number   (Without
Brackets) const char * myWriteAPIKey "VKEYU77SR4KGAG0L"; //Your Write
API Key

void setup()

{

Serial.begin(9600); delay(10);

WiFi.begin(ssid, password);

ThingSpeak.begin(client);

pinMode(16,INPUT);

}

int main (void)

{

uint8_t glu = 0x00;

bool up = true;

for(;;) {

int OCR0A = glu;

glu += up ? 1 : -1;

    if (glu == 0xff)

up = false;
```

```
else if (glu = =
   0x00) up = true;
delay(10);
}
}
void loop()
{
Int
 val=digitalRead(16);
 if(val==1)
{
ThingSpeak.writeField(myChannelNumber, 1,glu, myWriteAPIKey); delay(1000);
delay(1000);delay(1000);
}
else
{
ThingSpeak.writeField(myChannelNumber, 1,val, myWriteAPIKey); delay(1000);
}
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 20, 4);
#include "MAX30100_PulseOximeter.h"
#define REPORTING_PERIOD_MS     1000
const int analogInPin = A0;
PulseOximeter pox;
uint32_t tsLastReport = 0;
```

```
void setup()
{

  Serial.begin(9600);
lcd.init();
  lcd.backlight();
  lcd.setCursor(6,0);
  lcd.print("VENTILATOR");
   lcd.setCursor(4,1);
  lcd.print("USING ARDUINO");
     lcd.setCursor(6,2);
  lcd.print("AMBU BAG");
delay(2000);
pinMode(D5 , OUTPUT);

 if (!pox.begin())
    {

        for(;;);
    }
    else
    {

    }

     pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
    pox.setOnBeatDetectedCallback(onBeatDetected);

}
```

```
void loop()
{



  pox.update();
   if (millis() - tsLastReport > REPORTING_PERIOD_MS)
   {

int sensorValue = analogRead(A0);
    float voltage = sensorValue*(5.0/1023.0)*33;

lcd.clear();
lcd.setCursor(0,0);
lcd.print("HEART RATE:");
lcd.print(pox.getHeartRate());
lcd.setCursor(0,1);
lcd.print("SPO2:");
lcd.print(pox.getSpO2());
lcd.setCursor(0,2);
lcd.print("TEMPERATURE:");
lcd.print(voltage);

tsLastReport = millis();

if(pox.getHeartRate()>100)
{
digitalWrite(D5,HIGH);

}
```

```
else if(pox.getSpO2()>100)
{
digitalWrite(D5,HIGH);


}
else if(voltage>50)
{
digitalWrite(D5,HIGH);


}
```

ARTIFICIAL INTELLIGENCE PART:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
 import seaborn as sns
%matplotlib inline
 import warnings
```

```
warnings.filterwarnings('ignore')
from subprocess import check_output
data = pd.read_csv('C:\\Users\\DELL\\Dataset\\diabetes.csv')
data.head()
import pickle
data.describe()
    g = sns.PairGrid(data, vars=['Glucose', 'Insulin', 'BMI'], hue="Outcome",
     size=2.4)
g.map_diag(plt.hist)
g.map_upper(plt.scatter)
g.map_lower(sns.kdeplot, cmap="Blues_d")
g.add_legend()
plt.show()
g = sns.PairGrid(data, vars=['Age', 'SkinThickness', 'BloodPressure'],
hue="Outcome", size=2.4)
g.map_diag(plt.hist)
g.map_upper(plt.scatter)
g.map_lower(sns.kdeplot, cmap="Blues_d")
g.add_legend()
plt.show()
columns=['Glucose','Age',BloodPressure','Insulin','BMI','SkinThickness','Pregnanci
es', 'DiabetesPedigreeFunction']
 n_cols = 2
n_rows = 4
idx = 0
```

```
for i in range(n_rows):
fg,ax = plt.subplots (nrows=1,ncols=n_cols,sharey=True,figsize=(8, 2.4))
for j in range(n_cols):
    sns.violinplot (x = data.Outcome, y=data[columns[idx]], ax=ax[j])
    idx += 1
   if idx >= 8:
     break
max_skinthickness = data.SkinThickness.max()
data = data[data.SkinThickness!=max_skinthickness]
def replace_zero(df, field, target):
   mean_by_target = df.loc[df[field] != 0,[field,target]].groupby(target).mean()
    data.loc[(df[field] == 0) & (df[target] ==0),field]=mean_by_target.iloc[0][0]
    data.loc[(df[field] == 0) & (df[target]==1),field]=mean_by_target.iloc[1][0]

for col in ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']:
    replace_zero(data, col, 'Outcome')
data.describe()
X = data.iloc[:,:-1]
     y = data.iloc[:, -1]
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=100
)
print(X_train.shape)
print(X_test.shape)
print(y_train.size)
print(y_test.size)
```

```
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=8)
clf_ = knn.fit(X_train, y_train)
y_pred = clf_.predict(X_test)
print ('Accuracy is {}'.format(accuracy_score(y_test,y_pred ))) y_pred
model = pickle.load(open('model.pkl','rb'))
pickle.dump(knn,open('model.pkl','wb'))
    print(model.predict([[5,105,72,29,325,36.9,0.159,28]]))
```

REFERENCES

1. Abdul Mohsen Al Husseini1, Heon Julee, Justin Negrete1, Stephen Powelson, Amelia Servi,Alexander Slocum, Jussi Saukkonen(2010) - " Design and Prototyping Of a low cost portable ventilator",Massachusetts Institute of Technology, Department of Mechanical Engineering,Boston University, School of Medicine.

2. Aliaksei Petsiuk, Nagendra G. Tanikella, Samantha Dertinger, Adam Pringle ,Shane Oberloier a, Joshua M. Pearce(2020) -" Partially RepRapable Automated open source bag value mask based Ventilator"Department of Electrical & Computer Engineering, Michigan Technological University, USA

3. Md. Rakibul Islam, Mohiuddin Ahmad,Md. Shahin Hossain(2020)- "Designing an Electro Mechanical Ventilator Based on Double CAM Integration Mechanism", Khulna University of Engineering & technology

4. Rouf-ul-Aalam, Afshan Amin Khan, Dr Liyaqat Nazir (2022) -" Design of Efficient Low-cost Ventilator for Emergency COVID19 Patients", Department of ECE, IOT, University of Kashmir, Department of CSE,NIT Srinagar.

5. S.M.T. Hossain, M.R. Halder, A. Aman, M.R. Islam, M.T. Rahman(2018), "Design Construction and Performance Test of a Low-Cost Portable Mechanical Ventilator for, International Conference on Mechanical, Industrial and Energy Engineering".