ABSTRACT

Ventilators are medical equipment that helps those who are unable to breathe. It aids in supplying the lungs with air. Both in hospitals and ambulances, it will help save lives. People who need continuous ventilation can use their homes. There may be fewer amenities in ambulances. Most of the time, ambulances lack a basic first aid setup. Accidents cannot always be prevented, but simple equipment like the "Smart ventilator system" in ambulances can help save lives. It requires less labour and is rechargeable, portable, and affordable. For technicians, the system is simple to use. Patient's data are taken using biosensors like a pulse oximeter, ECG, temperature sensor, humidity sensor, and the data are sent to the concerned doctor through the cloud. In order to enable telemetric operation of the ventilator system, an application-oriented on/off control of the entire system is also enabled. Thus, the proposed system saves the lives of people with low cost.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE SYSTEM

Ventilators is a device that supports breathing process who are unable to breathe. It helps in pumping air into the lungs It will be helpful in solving lives both in hospitals and ambulances. People who require long term ventilation can be used at homes. Ambulances may have some less facilities. Accident cannot be avoided but some basic aids in ambulances like “Smart Ambu Bag” can save the life of the person. This Smart Ambu Bag is portable, rechargeable, low cost and uses less man power. It is a easy usable system for technicians. In most cases ambulances do not have basic first aid setup. This integrated system of ambu bag with biosensors like pulse oximeter, ECG, temperature and humidity sensor sends real time messages via cloud. An application oriented On/Off system is also enable so that , the smart ambu bag can be operated telemetrically.

1.2 OBJECTIVES OF THE PROJECT

1. To meet the increasing demand of mechanical ventilators due to COVID -19.
2. To provide low-cost alternative to ventilators.
3. To develop a very easy to operate system so that any less experience person must be able to operate it with ease.
4. To develop a cloud-based patient management system tohandle patients of COVID -19.

5. An ON/OFF system via mobile app so that can be operated from distance.

1.3 PROBLEM STATEMENT

Ventilators are one of the most important devices to keep COVID-19 patients in the most critical condition alive. As the global demand for ventilators is increasing and there is shortage of ventilators in our country as well, also managing patients during this time is a big task. Manual AMBU bags in ambulance which requires human effort and cannot be used for long time.

Two persons are needed for the aid of patients in critical condition. Accidents cannot be avoided but the lives of the people can be saved by effectively using this device. Good facility is not present in ambulance which can be implemented.

1.4 BENEFITS OF THE PROJECT

The benefits are:

- Patients at critical conditions can be given life support with this equipment.
- Death due to unavailability of ventilators can be reduced.
- Doctors can monitor the condition of patients 24/7.
- Amount of air output can be adjusted by tightening the straps.
- The prototype can be turned On/Off by mobile application.
- The buzzer system installed would alert when the patient’s pulse decreases.
- An abnormality in the ECG is also intimated through the mobile application.

1.5 PROJECT SCOPE/CONSTRAINTS

Ventilators are one of the most important devices to keep COVID-19 patients in the most critical condition alive. As the global demand for ventilators is increasing and there is shortage of ventilators in our country as well, also managing patients during this time is a big task, so the design of portable rechargeable battery operated Ambu bag compressing machine is been developed. The prototype model sends real time cloud messages to the doctors about the condition of the patient. It can be used for emergency purposes, in hospitals, Corona virus quarantine coaches, isolation wards and rural areas as well. The shortage of ventilators can be met effectively by developing this project on a larger scale. This project is a low-cost yet effective ventilating system for the people affected with COVID-19.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

We first tried to analyze those research work to find out best possible ways to implement our system on Smart Ambu bag. In research we found that this topic can be implemented with various biosensors and different setup have different each having a better function than one another.

In order to design, develop and to implement this system, several theoretical and technological surveys are made with previous papers. This review includes the technological development and control methods that are used in our project.

2.2 LITERATURE SURVEY

Thrupthi, A. R., et al. "Cost Effective Smart Ventilation System for COVID-19 Pandemic." International Journal of Research in Engineering, Science and Management 5.6 (2022): 116-119.

Medical term for artificial ventilator is electrical ventilator, in which an electrical motor replaces spontaneous breathing, which may necessitate the use of a ventilator. Manual compression of the Ambu bag can help with breathing. Engineering and architectures must be guided by current rules for the realization of every project. And we must check whether the parameters given is sufficient or not. From this cost-effective ventilator system comes into play. For treating covid-19 particularly in nations like India where we have large population there is a moment necessity of cost-effective ventilators which can operate with sufficient functionality to cure Covid patients. This paper describes the design of cost-effective ventilator system which detects oxygen level, body temperature and pulse rate.

Balamurugan C.R., Kasthuri A., Malathi E. Dharanidharan S., Hariharan D., Kishore B.V., Venkadesh T., (Design of Ventilator Using Arduino for Covid Pandemic) (01 April 2021).

This ventilator is made with push mechanism in each breath. This ventilator is very cheap and affordable. Motor mechanism is used push the air bag. When oxygen level count is low this mechanism is performed. Small screen is used to display the oxygen level at real time. The entire system is controlled with Arduino. If the oxygen level is low the buzzer is ring. Toggle are switching and variable pot to check the breath length and BPM level of patients.

Leonardo Acho, Alessandro N. Vargas, Gisela Pujol–Vázquez, (Low-Cost, Open Source Mechanical Ventilator with Pulmonary Monitoring for COVID-19 Patients) (12 September 2020)

This article shows that construction of low-cost, open-source mechanical ventilator. This article also shows that numerical method for monitoring patients pulmometry condition. With the help of pressure sensor, we can classify the patients are healthy or unhealthy lungs. An Arduino board collects the information from pressure sensor and sends them to raspberry pi. The raspberry pi commands the actutor and breathing bag compress accordingly. According to manufacture the pressure sensor can measure differentially pressure of up to 70 cm H2O.

the gear was attached to the servo meter rod. The rod was made by plexi glass bar. The radius of this gear is 2.5cm.

Aliaksei Petsiuk, Nagendra G. Tanikella, Samantha Dertinger, Adam Pringle, Shane Oberloier, Joshua M. Pearce, (Partially Rep Rapable automated open source bag valve mask-based ventilator)

This article shows that development of simple and easy to build portable automated mask value bag. This handle with Arduino controller with real time operating system installed on largely rep rap 3d printable parameter component based structure. For Arduino extensively grows the conceivable outcomes of the controller. A real-time operating system gives fundamental capacities to software tasks, like planning, dispatching, inter-task communication, and synchronization.

Couchman, B. A. et al. (Nurses role in prevention and management of mechanical ventilation related complications) (2006)

In their article titled, nursing care of the mechanically ventilated patient: What does the evidence say? summarized as nursing care and management of mechanically ventilated patient is challenging and requires nursing expertise for knowing the technological issues undying the patient cantered approach. Mechanical ventilation precipitates several actual and potential complications for the critically ill clients. Use of ventilator care is effective in mechanically ventilated patients producing positive outcomes which consists of four interventions, those are; elevation of head of the bed, sedation vacation, peptic ulcer prophylaxis and deep vein thrombosis prophylaxis. The nursing care practice lacks in supportive significant evidence for proving one care approach is better than the other. In the care of mechanically ventilated patient the best nursing care practice is the use of evidence-based practice conjunction with comprehensive and systematic patient.

Vikky Bhakre*, Ajay Pathrabe, Journal of Optoelectronics and CommunicationVolume 3 Issue 2, HBRP Publication Page 1-5 (2021) Journal

DIY Ventilator using Arduino with blood oxygen sensing for covid pandemic. ventilator bag driven by Direct Current motors, 2 side push mechanism to push the air bag. Toggle for switching, variable pot to check the breath length, BPM levels of the patient. There is a lot of future work to be upgraded to make it medical grade hardware. The team of engineers and clinicians designed and tested an emergency ventilator that uses a single limb portable ventilator circuit. Ventilator performance was assessed using an Ingmar ASL 5000 breathing simulator, and it was compared with a commercially available mechanical ventilator. The emergency ventilator provides volume control mode, intermittent mandatory ventilation and continuous positive airway pressure.

CHAPTER 3

PROPOSED SYSTEM

3.1 AMBU BAG TECHNOLOGY

A bag valve mask (BVM), 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. Thedevice is a required part of respiration kits for trained professionals in out- of-hospital settings (such aa ambulance crews) and is also frequently usedin hospitals as part of standard equipment found on a crash cart, in emergency rooms or other critical care settings.

3.2 METHOD USED FOR AMBU ABG

Manual resuscitators cause the gas inside the inflatable bag portion to be force-fed to the patient via a one-way valve when compressed by the rescuer; the gas is then ideally delivered through a mask and into the patient's trachea, bronchus and into the lungs. In order to be effective, a bag valve mask must deliver between 500 and 600 mm of air to a normal male adult patient's lungs, but if supplemental oxygen is provided 400 ml may still be adequate. Squeezing the bag once every 5 to 6 seconds for an adult oronce every 3 seconds for an infant or child provides an adequate respiratory rate (10–12 respirations per minute in an adult and 20 per minute in a child or infant)

3.3 METHODOLOGY

EXISTING SYSTEM

The most pressing shortages facing hospitals during the COVID-19 emergency is a lack of ventilators. These machines can keep patients breathing when they no longer can on their own and they can cost around $30,000 each. Design and development of a low-cost portable ventilator could be a possible way, that can help pneumonia cases of COVID-19 patient.

PROPOSED SYSTEM

Proposed low-cost ventilator delivers breaths by compressing a conventional Bag Valve Mask (BVM) or Ambu bag with a pivoting motor drive mechanism, eliminating the need for a human operator

Proposed low-cost ventilator delivers breaths by compressing a conventional Bag Valve Mask (BVM) or Ambu bag with a pivoting motor drive mechanism, eliminating the need for a human operator.

Among other features, the machine should have invasive and non-invasive feature, and supports 500-600 mL tidal volume, with a continuous working capability for several days. Based on calculation, 12Respiratory rate (RR)/min can provide required amount of tidal volume to the pneumonia patient.

Strategy of automatic arm actuated BVM compression is proven to be a viable option to achieve low-cost, low-power and portable ventilator technology that provides essential ventilator features at a fraction of the cost of existing models.

CHAPTER 4

HARDWARE DESCRIPTION

4.1 BAG VALVE MASK (BVM) OR AMBU BAG

Used for resuscitation in emergency situations, available at local chemists.

- Compressible self-refilling ventilation bag with non-re breathing patient valve with pressure limitation valve to minimize the risk of overinflating.
- 100% latex free.

Fig.4.1 Ambu bag

4.2 RESPIRATORY MASK

An Respiratory mask provides a method to transfer breathing oxygen gas from a storage tank to the lungs. Oxygen masks may cover only the nose and mouth (oral nasal mask) or the entire face (full-face mask). They may be made of plastic, silicone, or rubber. In certain circumstances, oxygen may be delivered via a nasal cannula instead of a mask. Medical plastic oxygen masks are used primarily by medical care providers for oxygen therapy because they are disposable and so reduce cleaning costs and infection risks. Mask design can determine accuracy of oxygen delivered with many various medical situations requiring treatment with oxygen. Oxygen is naturally occurring in room air at 21% and higher percentages are often essential in medical treatment. Oxygen in these higher percentages is classified as a drug with too much oxygen being potentially harmful to a patient’s health, resulting in oxygen dependence over time, and in extreme circumstances patient blindness. For these reasons oxygen therapy is closely monitored. Masks are light in weight and attached using an elasticated headband or ear loops. They are transparent for allowing the face to be visible for patient assessment by healthcare providers, and reducing a sensation of claustrophobia experienced by some patients when wearing an oxygen mask.

Fig.4.2 Respiratory mask

The vast majority of patients having an operation will at some stage wear an oxygen mask; they may alternatively wear a nasal cannula but oxygen delivered in this way is less accurate and restricted in concentration. Silicone and rubber oxygen masks are heavier than plastic masks. They are designed to provide a good seal for long-duration use by aviators, medical research subjects, and hyperbaric chamber and other patients who require administration of pure oxygen, such as carbon monoxide poisoning and decompression sickness victims. Dr. Arthur H. Bulbulian pioneered the first modern viable oxygen mask, worn by World War II pilots and used by hospitals. Valves inside these tight-fitting masks control the flow of gases into and out of the masks, so that rebreathing of exhaled gas is 12inimized.Hoses or tubing connect an oxygen mask to the oxygen supply. Hoses are larger in diameter than tubing and can allow greateroxygen flow. When a hose is used it may have a ribbed or corrugated design to allow bending of the hose while preventing twisting and cutting off the oxygen flow. The quantity of oxygen delivered from the storage tank to the oxygen mask is controlled by a valve called a regulator. Some types of oxygenmasks have a breathing bag made of plastic or rubber attached to the mask oroxygen supply hose to store a supply of oxygen to allow deep breathing without waste of oxygen with use of simple fixed flow regulators.

4.3 DC WIPER MOTOR

For this prototype used 12V DC gear motor of 7 kg/cm torque capacity. This has the range of 6 – 700rpm which is suitable for the project

Fig.4.3 DC Wiper motor

4.4 BATTERY

In our project we are using secondary type battery. It is rechargeable type. A battery is one or more electrochemical cells, which store chemical energy and make it available as electric current. There are two types of batteries, primary (disposable) and secondary (rechargeable), both of which convert chemical energy to electrical energy. Primary batteries can only be used once because they use up their chemicals in an irreversible reaction. Secondary batteries can be recharged because the chemical reactions they use are reversible; they are recharged by running a charging current through the battery, but in the opposite direction of the discharge current. Secondary, also called rechargeable batteries can be charged and discharged many times before wearing out. After wearing out some batteries can be recycled. Batteries have gained popularity as they became portable and useful for manypurposes. The use of batteries has created many environmental concerns, suchas toxic metal pollution. A battery is a device that converts chemical energy directly to electrical energy it consists of one or more voltaic cells. Each voltaic cell consists of two half cells connected in series by a conductive electrolyte. One half-cell is the positive electrode, and the other is the negativeelectrode. The electrodes do not touch each other but are electrically connected by the electrolyte, which can be either solid or liquid. A battery canbe simply modeled as a perfect voltage source which has its own resistance, the resulting voltage across the load depends on the ratio of the battery's internal resistance to the resistance of the load.

When the battery is fresh, its internal resistance is low, so the voltage across the load is almost equal to that of the battery's internal voltage source. As the battery runs down and its internal resistance increases, the voltage drop across its internal resistance increases, so the voltage at its terminals decreases, and the battery's ability to deliver power to the load decreases

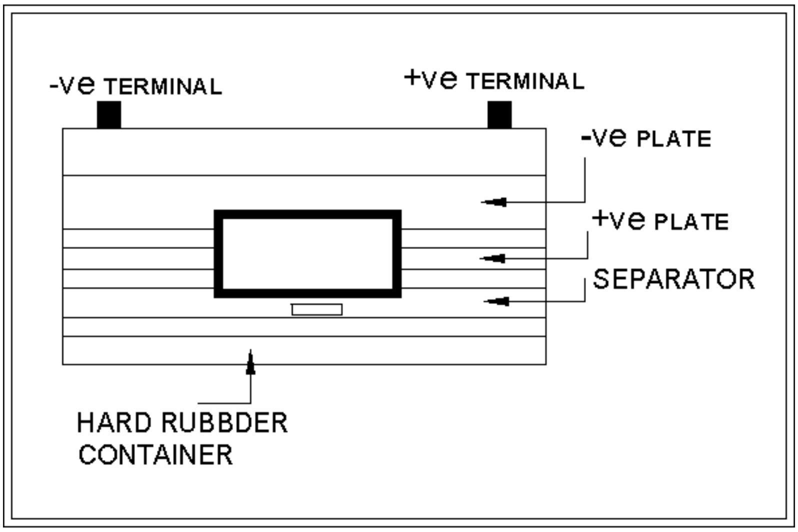

Fig.4.4 Battery

4.5 ESP8266 Wi-Fi MODULE

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution.

The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic control monitor. Please do not power it directly from your 5V dev board.

This new version of the ESP8266 Wi-Fi Module has increased the flash disk size from 512k to 1MB

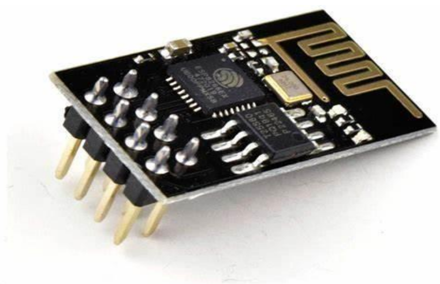

Fig.4.5 ESP8266 Wi-Fi Module

4.6 RELAY

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off. So relays have two switch positions and they are double throw (changeover) switches. Relays allow one circuit to switch a second circuit which can be completely separate from the first. The link is magnetic and mechanical. The coil of a relay passes a relatively large current, typically 30mA for a 12V relay, but it can be as much as 100mA for relays designed to operate from lower voltages. Most ICs (chips) cannot provide this current and a transistor is usually used to amplify the small IC current to the larger value required for the relay coil. The maximum output current for the popular 555 timer IC is 200mA so these devices can supply relay coils directly without amplification.

Relays are usually SPDT or DPDT but they can have many more sets of switch contacts, for example relays with 4 sets of changeover contacts are readily available. Most relays are designed for PCB mounting but you can solder wires directly to the pins providing you take care to avoid melting the plastic case of the relay. The animated picture shows a working relay with its coil and switch contacts. You can see a lever on the left being attracted by magnetism when the coil is switched on. This lever moves the switch contacts. There is one set of contacts (SPDT) in the foreground and another behind them, making the relay DPDT.

Fig.4.6 Relay

4.7 ARDUINO UNO

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

- The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features:
- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Fig.4.7 Arduino UNO

4.8 LCD DISPLAY

At the heart of the adapter is an 8-Bit I/O Expander chip – PCF8574. This chip converts the I2C data from an Arduino into the parallel data required by the LCD display. The board also comes with a small tripod to make fine adjustments to the contrast of the display. In addition, there is a jumper on the board that supplies power to the backlight. To control the intensity of the backlight, you can remove the jumper and apply an external voltage to the header pin that is marked as ‘LED’.

Fig.4.8 LCD Display

4.9 IC 7805 VOLTAGE REGULATOR

It is a three-pin IC; input pin for accepting incoming DC voltage, ground pin for establishing ground for the regulator, and output pin that supplies the positive 5 volts. Input Voltage: Minimum 7V, Maximum 25V Output Current: 1.5A Operating Virtual Junction Temperature: Minimum 0, Maximum 125°C Possible High Temperatures If differences between the input and output voltages are not well managed, LM7805 can overheat, whichmay result in malfunctioning.

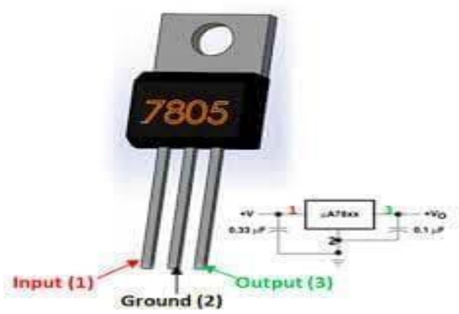

Fig.4.9 IC 7508 Voltage Regulator

Solutions Include: Limiting input voltage to 2-3 volts above the output regulated voltage.

4.10 ECG SENSOR

This sensor is a cost-effective board used to measure the electrical activity of the heart. This electrical activity can be charted as an ECG or Electrocardiogram and output as an analog reading. ECGs can be extremely noisy, the AD8232 Single Lead Heart Rate Monitor acts as an op-amp to help obtain a clear signal from the PR and QT Intervals easily

The AD8232 is an integrated signal conditioning block for ECG and other biopotential measurement applications. It is designed to extract, amplify, and filter small biopotential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement.

The AD8232 module breaks out nine connections from the IC that you can solder pins, wires, or other connectors to. SDN, LO+, LO-, OUTPUT, 3.3V, GND provide essential pins for operating this monitor with an Arduino or other development board. Also provided on this board are RA (Right Arm), LA (Left Arm), and RL (Right Leg) pins to attach and use your own custom sensors. Additionally, there is an LED indicator light that will pulsate to the rhythm of a heartbeat.

Fig.4.10 ECG Sensor

4.11 DHT11 TEMPERATURE SENSOR

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc. to measure humidity and temperature instantaneously.

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1 Hz. i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

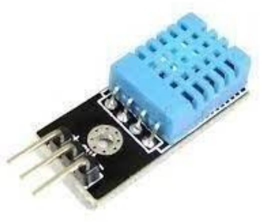

Fig.4.11 DHT11 Sensor

4.12 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.

The pin configuration of the buzzer is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-' symbol or short terminal and it is connected to the GND terminal.

Fig.4.12 Buzzer

4.13 SWITCH

A switch is an electrical component that can disconnect or connect the conducting path in an electrical circuit, interrupting the electric current or diverting it from one conductor to another.The most common type of switch is an electromechanical device consisting of one or more sets of movable electrical contacts connected to external circuits. When a pair of contacts is touching current can pass between them, while when the contacts are separated no current can flow.

Fig.4.13 Switch

4.14 STEEL FRAME

Steel is an alloy made up of iron with typically a few tenths of a percent of carbon to improve its strength and fracture resistance compared to other forms of iron. Many other elements may be present or added. Stainless steels that are corrosion- and oxidation-resistant need typically an additional 11% chromium. Because of its high tensile strength and low cost, steel is used in buildings, infrastructure, tools, ships, trains, cars, machines, electrical appliances, weapons, and rockets. Iron is the base metal of steel. Depending on the temperature, it can take two crystalline forms (allotropic forms): body-centered cubic and face-centered cubic. The interaction of the allotropes ofiron with the alloying elements, primarily carbon, gives steel and cast iron their range of unique properties.

In pure iron, the crystal structure has relatively little resistance to the iron atoms slipping past one another, and so pure iron is quite ductile, or soft and easily formed. In steel, small amounts of carbon, other elements, and inclusions within the iron act as hardening agents that prevent the movement of dislocations. The carbon in typical steel alloys may contribute up to 2.14% of its weight. Varying the amount of carbon and many other alloying elements, as well as controlling their chemical and physical makeup in the final steel (either as solute elements, or as precipitated phases), impedes the movement of the dislocations that make pure iron ductile, and thus controls and enhances its qualities. These qualities include the hardness, quenching behaviour, need for annealing, tempering behaviour, yield strength, and tensile strength of the resulting steel. The increase in steel's strength compared to pure iron is possible only by reducing iron's ductility.

Steel was produced in bloomery furnaces for thousands of years, but its large-scale, industrial use began only after more efficient production methods weredevised in the 17th century, with the introduction of the blast furnace and production of crucible steel. This was followed by the open-hearth furnace and then the Bessemer process in England in the mid-19th century. With the invention of the Bessemer process, a new era of mass-produced steel began. Mild steel replaced wrought iron. The German states saw major steel prowess over Europe in the 19th century.

Further refinements in the process, such as basic oxygen steelmaking (BOS),largely replaced earlier methods by further lowering the cost of production and increasing the quality of the final product. Today, steel is one of the mostcommonly manufactured materials in the world, with more than 1.6 billion tons produced annually. Modern steel is generally identified by various gradesdefined by assorted standards organizations. The modern steel industry is oneof the largest manufacturing industries in the world, but is one of the most energy and greenhouse gas emission intense industries, contributing 8 % of global emissions.[2] However, steel is also very reusable: Steel is one of the world's most-recycled materials, with a recycling rate of over 60% globally.

Fig.4.14 Steel Fame

4.15 WELDING

Welding is a fabrication process that joins materials, usually metals or thermoplastics, by using high heat to melt the parts togetherand allowing them to cool, causing fusion. Welding is distinct from lower temperature techniques such as brazing and soldering, which do not melt the base metal (parent metal).

In addition to melting the base metal, a filler material is typically added to the joint to form a pool of molten material (the weld pool) that cools to for joint that, based on weld configuration (butt, full penetration, fillet, etc.), can be stronger than the base material. Pressure may also be used in conjunction with heat or by itself to produce a weld. Welding also requires a form of shield to protect the filler metals or melted metals from being contaminated or oxidized.

Many different energy sources can be used for welding, including a gas flame (chemical), an electric arc (electrical), a laser, an electron beam, friction, and ultrasound. While often an industrial process, welding maybe performed in many different environments, including in open air, under water, and in outer space. Welding is a hazardous undertaking and precautionsare required to avoid burns, electric shock, vision damage, inhalation of poisonous gases and fumes, and exposure to intense ultraviolet radiation.

Until the end of the 19th century, the only welding process was forge welding, which blacksmiths had used for millennia to join iron and steel by heating and hammering. Arc welding and oxy-fuel welding were among the first processes to develop late in the century, and electric resistance welding followed soon after. Welding technology advanced quickly during the early 20th century as world wars drove the demand for reliable and inexpensive joining methods. Following the wars, several modern welding techniques were developed, including manual methods like shielded metal arcwelding, now one of the most popular welding methods, as well as semi- automatic and automatic processes such as gas metal arc welding, submergedarc welding, flux-cored arc welding and electroslag welding. Developments continued with the invention of laser beam welding, electron beam welding, magnetic pulse welding, and friction stir welding in the latter half ofthe century. Today, as the science continues to advance, robot welding is commonplace in industrial settings, and researcher continue to develop new welding methods and gain greater understanding of weld quality.

Fig.4.15 Welding

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 BLYNK APPLICATION

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things.

There are three major components in the platform:

- **Blynk App** - allows to you create amazing interfaces for your projects using various widgets we provide.
- **Blynk Server** - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your Private Blynk server locally. It’s open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.
- **Blynk Libraries** - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

Now imagine: every time you press a Button in the Blynk app, the message travels to space the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a blynk of an eye.

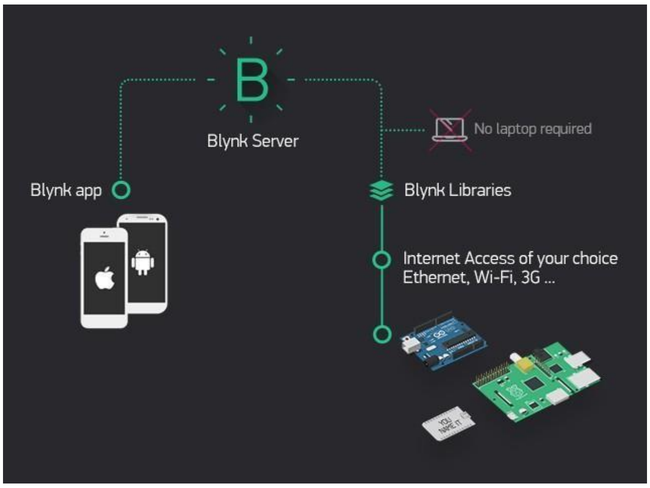

Fig.5.1 Process of interfacing blynk app

Features

Similar API & UI for all supported hardware & devices Connection to the cloud using:

- ✓ Wi-Fi
- ✓ Bluetooth and BLE
- ✓ Ethernet
- ✓ USB (Serial)
- ✓ GSM

- ➢ Set of easy-to-use Widgets
- ➢ Direct pin manipulation with no code writing
- ➢ Easy to integrate and add new functionality using virtual pins
- ➢ History data monitoring via Super Chart widget
- ➢ Device-to-Device communication using Bridge Widget
- ➢ Sending emails, tweets, push notifications, etc.
- ➢ new features are constantly added!

You can find example sketches covering basic Blynk Features. They are included in the library. All the sketches are designed to be easily combined with each other.

What do I need to Blynk?

***1.* Hardware**.

An Arduino, Raspberry Pi, or a similar development kit. **Blynk works over the Internet.** This means that the hardware you choose should be able to connect to the internet. Some of the boards, like Arduino Uno will need an Ethernet or Wi-Fi Shield to communicate, others are already Internet-enabled: like the ESP8266, Raspberry Pi with Wi-Fi dongle, Particle Photon or SparkFun Blynk Board. But even if you don't have a shield, you can connect it over USB to your laptop or desktop (it's a bit more complicated for newbies, but we got you covered). What's cool, is that the list of hardware that works with Blynk is huge and will keep on growing.

***2.* A Smartphone**.

The Blynk App is a well designed interface builder. It works on both iOS and Android.

Getting Started With The Blynk App

1. Create a Blynk Account

After you download the Blynk App, you'll need to create a New Blynk account. This account is separate from the accounts used for the Blynk Forums, in case you already have one.

We recommend using a **real** email address because it will simplify things later.

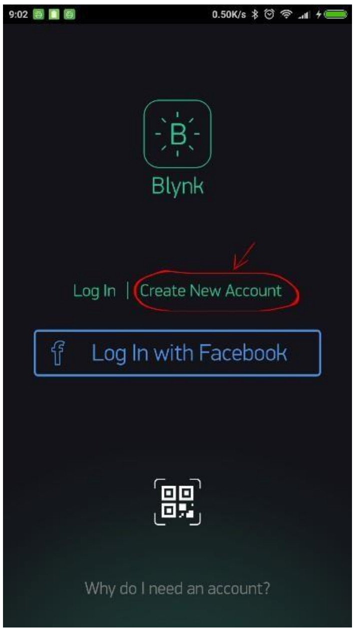

Fig.5.2 Logging in to blynk application

Why do I need to create an account?

An account is needed to save your projects and have access to them from multiple devices from anywhere. It's also a security measure. You can always set up your own Private Blynk Server and have full control

2. Create a New Project

After you've successfully logged into your account, start by creating a new project.

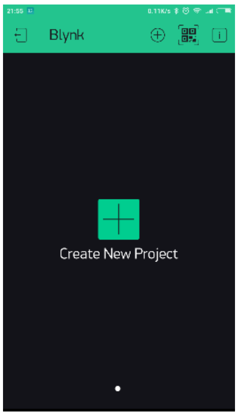

Fig.5.3 Creating new project

3. Choose Your Hardware

Select the hardware model you will use. Check out the list of supported hardware!

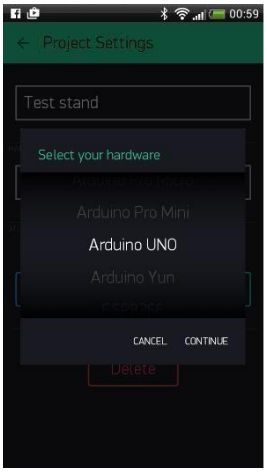

Fig.5.4 Interfacing hardware

4. Auth Token

Auth Token is a unique identifier which is needed to connect your hardware to your smartphone. Every new project you create will have its own Auth Token. You’ll get Auth Token automatically on your email after project creation. You can also copy it manually. Click on devices section and selected required device

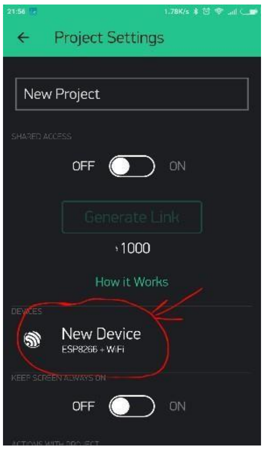

Fig.5.5 Interfacing new device

And you'll see token

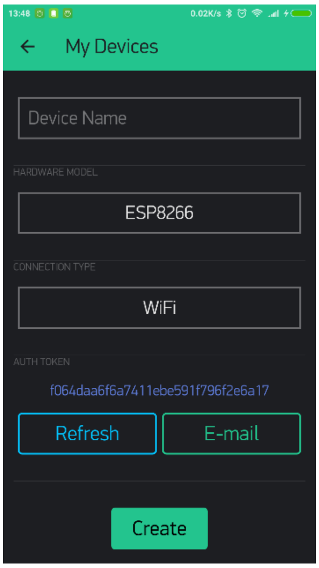

Fig.5.6 Adding Device name

Don't share your Auth Token with anyone, unless you want someone to have access to your hardware. It's very convenient to send it over e-mail. Press the e-mail button and the token will be sent to the e-mail address you used for registration. You can also tap on the Token line and it will be copied to the clipboard.

Now press the **"Create"** button.

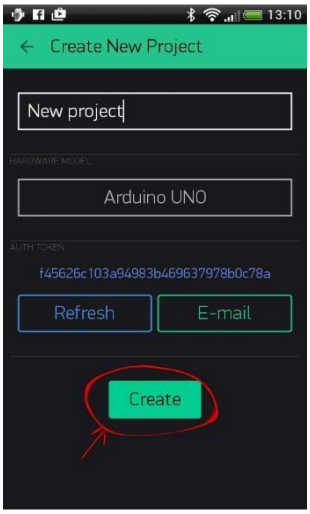

Fig.5.7 Adding project name

Add a Widget

Your project canvas is empty, let’s add a button to control our LED. Tap anywhere on the canvas to open the widget box. All the available widgets are located here. Now pick a button

Widget Box

Fig.5.8 Widget Box

Drag-n-Drop - Tap and hold the Widget to drag it to the new position. **Widget Settings -** Each Widget has it’s own settings. Tap on the widget to get to them.

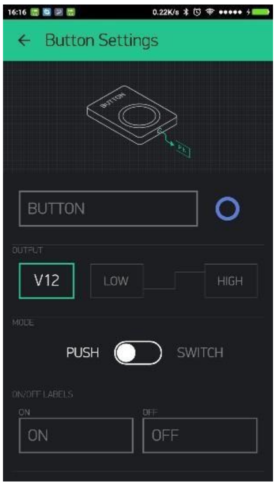

Fig.5.9 Application settings

The most important parameter to set is **PIN**

5.2 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programsto Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supportsthe languages C and C++ using special rules of code structuring.

The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution

Fig.5.10 Arduino IDE

5.3 IOT PAGE

- The Internet of Things, or IoT, refers to the billions of physical devices around the world that are now connected to the internet, all collecting and sharing data.
- Thanks to the arrival of super-cheap computer chips and the ubiquity of wireless networks, it's possible to turn anything, from something as small as a pill to something as big as an aero plane, into a part of the IoT.
- Connecting up all these different objects and adding sensors to them adds a level of digital intelligence to devices that would be otherwise dumb, enabling them to communicate real-time data without involving a human being.

CHAPTER 6

PROPOSED METHODOLOGY

6.1 METHODOLOGY

Proposed low-cost ventilator delivers breaths by compressing a conventional Bag Valve Mask (BVM) or Ambu bag with a pivoting motor drive mechanism, eliminating the need for a human operator. Proposed low-cost ventilator delivers breaths by compressing a conventional Bag Valve Mask (BVM) or Ambu bag with a pivoting motor drive mechanism, eliminating the need for a human operator. Among other features, the machine should have invasive and non-invasive feature, and supports 500-600 mL tidal volume, with a continuous working capability for several days. Based on calculation, 12Respiratory rate (RR)/min can provide required amount of tidal volume to the pneumonia patient. Strategy of automatic arm actuated BVM compression is proven to be a viable option to achieve low-cost, low-power and portable ventilator technology that provides essential ventilator features at a fraction of the cost of existing models.

6.2 BLOCK DIAGRAM

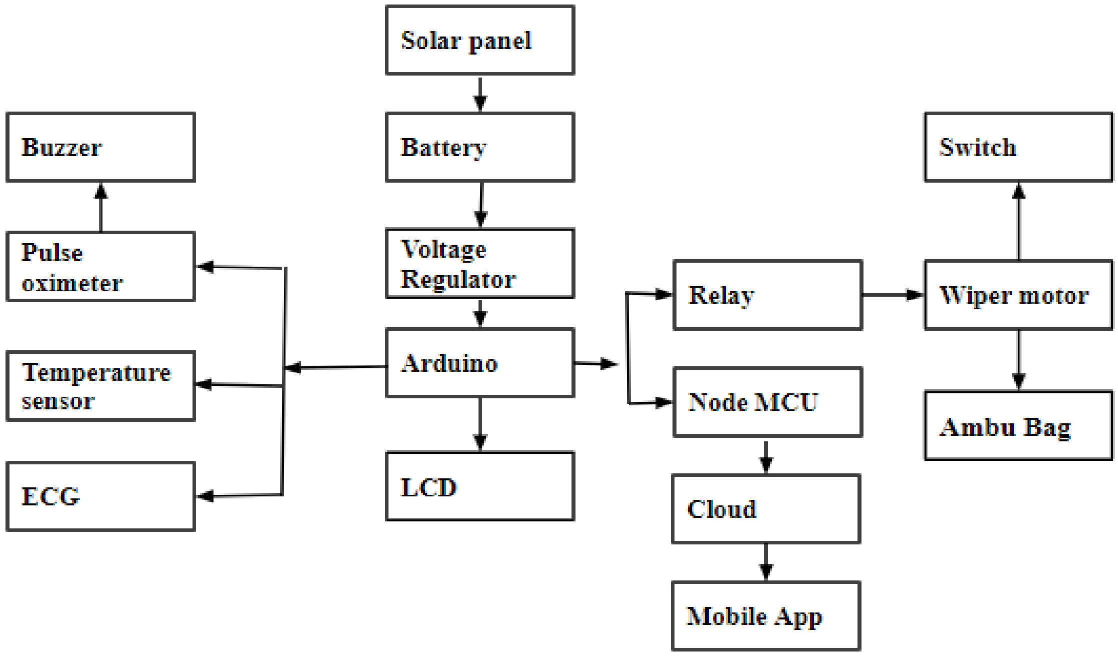

Fig.6.1 Block Diagram

CHAPTER 7

DESIGN AND ALGORITHM

7.1 FLOW CHART

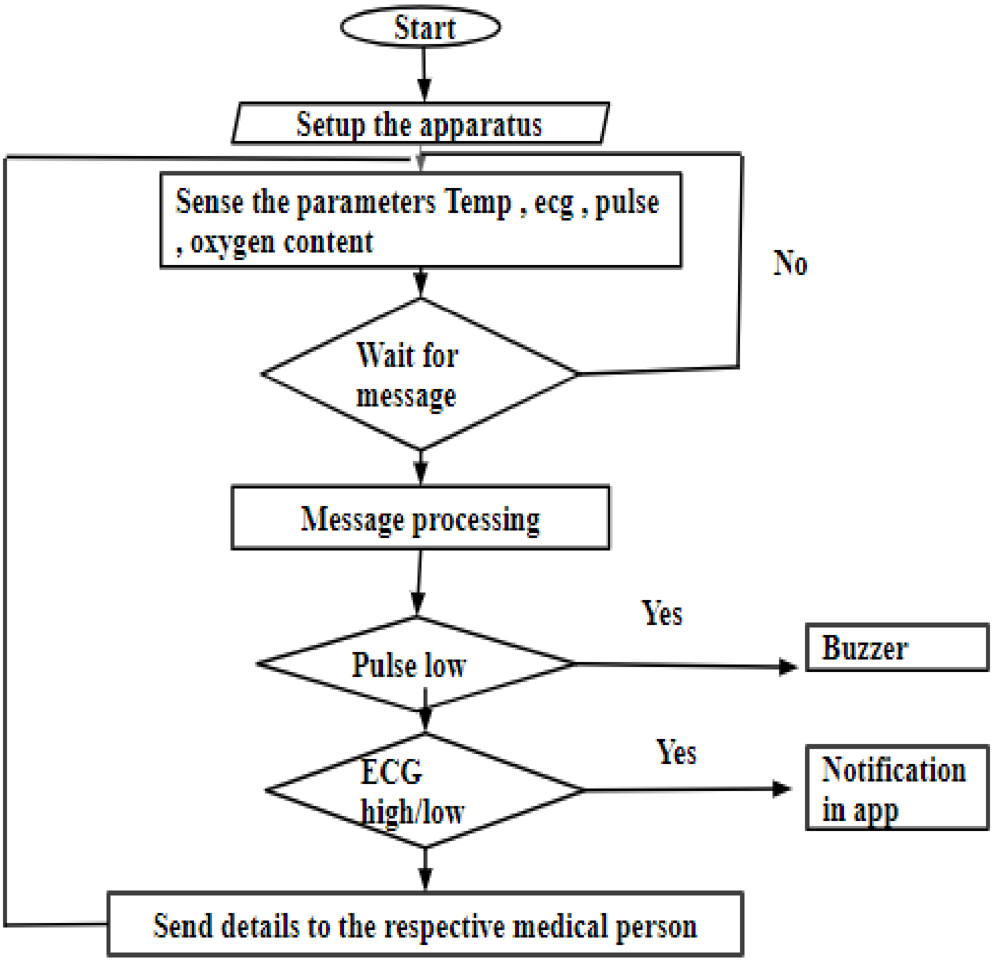

Fig.7.1 Flow Chart

7.2 ALGORITHM

STEP 1: Start

STEP 2: Establish the connection.

STEP 3: Monitor patient's heartrate, blood oxygen saturation, temperatureand ECG.

STEP 4: Check if it is normal.

STEP 5: If the pulse rate is low, buzzer is ON.

STEP 6: If the ECG values is abnormal, notification is sent in app.

STEP 7: The results are monitored on LCD.

STEP 8: Turn on relay

STEP 9: End

CHAPTER 8

CONCLUSION

8.1 RESULT

Arduino was programmed successfully using python with open CV and Micro controller was programmed successfully using Embedded C in HP vision camera to achieve face recognition door unlocking system. The individuals who were authorized to access the door he stands in front of the camera and it will capture the image of that person and compare with the data base images. If the images are recognized then automatically door will unlock. When a person image is not detected then it will a sent an email to the owner. Whereas when recognition of images, the correct pin was detected, relay released the door and access was granted for that person.

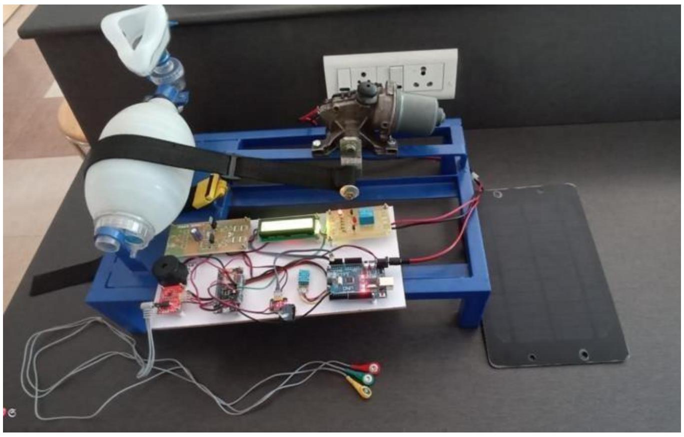

Fig.8.1 Experimental setup

Fig.8.2 Set up in Off state

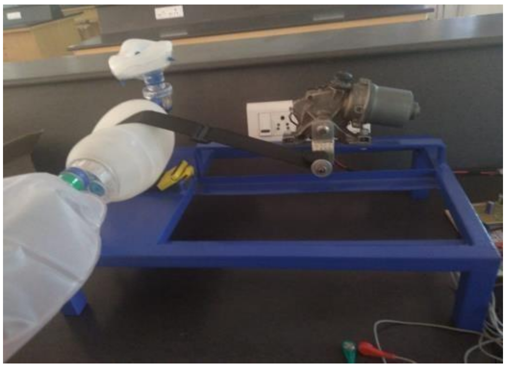

Fig.8.3 Set up in On state

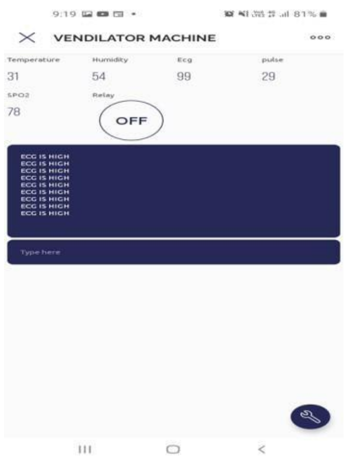

Fig.8.4 Real Time Monitoring Via App

Designed and developed a prototype of automatic ventilator to support
pneumonia cases for COVID-19 patients.

- It has a controlled breath rate of 12 RR/min and 500-600 mL tidal volume. It features assist control and provide a constant air flow to the lungs.
- Power requirement is very low and running for 3.5 hours on one battery charge at its most demanding setting. Battery backup also need to be checked.
- It is low cost, portable, light weight and able to run 2 DC motors at the same time.
- Further development includes pressure sensor, air flow meter, over pressure alarm and other safety features to make it more user friendly

8.2 CONCLUSION

- In this project, a prototype device to assist the patients who can partially breathe by their own is developed.
- This device is provided with very basic design and reliable structure that is easily acceptable by the patient.
- Main focus in this project is to minimize the components and increase the efficiency of the device, so that while using this device to the patient, they should feel as comfortable as the normal ventilator.
- In this project needle valve is used along with the potentiometer for replacing the flow analyzer so that the entire setup is cost effective.

- Arduino UNO board is used because it is easy to program. This research has led to the development of lab model ventilator.

8.3 FUTURE SCOPE

1. If the blood oxygen level low, then it automatically calls the doctor or ambulance.
2. And the particular location will be shared through GSM using GPS unit.

REFERENCES

[1] A. B. Baker, "Artificial respiration, the history of an idea," *Medical History*, vol. 15, no. 4, pp. 336–351, 1971

[2] A. C. F. d. Godoy, R. J. Vieira, and R. J. Vieira Neto, "Fluxo de saída de oxigênio fornecido por reanimadores manuais com balão auto-inflável em pacientes com ventilação espontânea," *Jornal Brasileiro de Pneumologia*, vol. 34, no. 4, pp. 212–216, 2008.

[3] A. Sing and O. P. Dadhich, "An insight of physiology of respiration in ayurveda, int," *Ayurvedic Medical Journal*, vol. 3, no. 11, p. 91, 2015.

[4] Bag Valve Mask (Bvm), "Firefighter Medic," http://www.firefightermedic.com/wiki/bag-valve-mask-bvm/.

[5] C. P. F. O'Donnell, A. T. Gibson, and P. G. Davis, "Pinching, electrocution, ravens' beaks, and positive pressure ventilation: a brief history of neonatal resuscitation," *Archives of Disease in Childhood Fetal and Neonatal Edition*, vol. 91, no. 5, pp. F369–F373, 2006.

[6] J. A. Cooper, J. D. Cooper, and J. M. Cooper, "Cardiopulmonary resuscitation," *Circulation*, vol. 114, no. 25, pp. 2839–2849, 2006.

[7] M. S. Eisenberg, *Life in the Balance: Emergency Medicine and the Quest to Reverse Sudden Death*, Oxford University Press, Oxford, UK, 1997.

[8] P. Baskett , "Peter J. Safar," *Resuscitation*, vol. 59, no. 1, pp. 3–5, 2003

[9] P. Brain and Galen, *Galen on Bloodletting: A Study of the Origins, Development, and Validity of His Opinions, with a Translation of the Three Works*, Cambridge University Press, Cambridge, UK, 1986.

[10] P. Cullen, “Self-inflating ventilation bags,” *Anesthesia & Intensive Care*, vol. 29, no. 2, p. 203, 2001.

[11] P. M. Fenton and G. Bell, “The patient inflating valve in anaesthesia and resuscitation breathing systems,” *Anaesthesia & Intensive Care*, vol. 41, no. 2, pp. 163–174, 2013.

[12] R. V. Rabinovitch, “History of mouth-to-mouth rescue breathing. Part 2: the 18th century,” *Critical care and resuscitation: Journal of the Australasian Academy of Critical Care Medicine*, vol. 8, no. 2, pp. 157–171, 2006.

[13] S. Payne, R. Tully, and R. Eltringham, “A new valve for draw-over anaesthesia,” *Anaesthesia*, vol. 65, no. 11, pp. 1080–1084, 2010.

[14] S. S. Batah and A. T. Fabro, “Pulmonary pathology of ARDS in COVID-19: a pathological review for clinicians,” *Respiratory Medicine*, vol. 176, 2021.

APPENDIX

PROGRAM:

```
//covid1973@gotgel.org
#include <EveryTimer.h>

#define PERIOD_MS 1000

EveryTimer timer;
bool active = true;

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define BLYNK_TEMPLATE_ID "TMPL3vGMIvu2L"
#define BLYNK_TEMPLATE_NAME "VENDILATOR MACHINE"
#define BLYNK_AUTH_TOKEN
"wnKR3UcFKJJD0BaxJbjDWPf3KCsKjzub"

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "IOT";
char pass[] = "123456789";
char auth[] = BLYNK_AUTH_TOKEN;
unsigned int m=0,act=0,val,val1,val2,val3,val4,val5;
```

```
#define relay  D5
String inputString = "";
unsigned char a[200];

#define swi D6
#define buzzer D0
int relaystat;

int  sec;
void setup()
{
  Serial.begin(9600);
  pinMode(swi,INPUT_PULLUP);
  pinMode(buzzer,OUTPUT);
  digitalWrite(buzzer,LOW);
  pinMode(relay,OUTPUT);
  digitalWrite(relay,HIGH);
  timer.Every(PERIOD_MS, action);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

}
  BLYNK_WRITE(V5)
{
  int button = param.asInt(); // read button
  if (button == 1)
  {
    digitalWrite(relay,LOW);
```

```
    relaystat=1;
  }
  else
  {
     digitalWrite(relay,HIGH);
     relaystat=0;
  }
}
void loop()
{
   timer.Update();
   Serial.print("SEc :");
   Serial.println(sec);

   while(Serial.available())
    {
        char data;
        data=Serial.read();

        a[m]=data;
          if(a[0] == '*')
          {
            if(m<=15)
            {m++;}
          }
      }
      if(m > 1)
```

```
{
  val = (a[1]-0x30)*100 + (a[2]-0x30)*10  + (a[3] - 0x30);
  val1 = (a[4]-0x30)*100 + (a[5]-0x30)*10  + (a[6] - 0x30);
  val2 = (a[7]-0x30)*100 + (a[8]-0x30)*10  + (a[9] - 0x30);
  val3 = (a[10]-0x30)*100 + (a[11]-0x30)*10  + (a[12] - 0x30);
  val4 = (a[13]-0x30)*100 + (a[14]-0x30)*10  + (a[15] - 0x30);

  m=0;
}


if(digitalRead(swi)==HIGH){digitalWrite(relay,LOW);
Serial.println("relay on by swithc");}
else if(relaystat==1){digitalWrite(relay,LOW);Serial.println("relay on
by IOT");}
else {digitalWrite(relay,HIGH); Serial.println("relay offed");}

if(sec>=10)
{
 if(val4<=60    &&    val4>=20){digitalWrite(buzzer,HIGH);    sec=0;
Serial.println("buzzer oned"); delay(2000); digitalWrite(buzzer,LOW);}
 else {digitalWrite(buzzer,LOW);Serial.println("buzzer offed");}
 sec=0;
}

 Blynk.virtualWrite(V0,val);
```

```
      delay(100);
      Blynk.virtualWrite(V1,val1);

      delay(100);



      Blynk.virtualWrite(V2,val2);
      delay(100);

      Blynk.virtualWrite(V3,val3);
      delay(100);

      Blynk.virtualWrite(V4,val4);
      delay(100);
      if(val2>=100){Blynk.virtualWrite(V6,"ECG IS HIGH");}

      Blynk.run();
      delay(100);
}



// Callback called by timer
void action()
{
  sec++;
}
```

```
#include <Wire.h>
#include <LCD_I2C.h>
LCD_I2C lcd(0x27);

#include <Wire.h>
#include "MAX30105.h"

#include "heartRate.h"

MAX30105 particleSensor;

const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred

float beatsPerMinute;
int beatAvg;

#include "DHT.h"
#define DHTPIN A0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
int h,t;
int bpm1 ,spo,ecg;

#define ecgv A3
```

```
#define ecg2 2
#define ecg3 3

int oxygenval;
void setup() {
  Serial.begin(9600);
dht.begin();

Wire.begin(); // gpio 2 and gpio 0 which are D4, and D3
pinMode(ecg2, INPUT); // Setup for leads off detection LO +
pinMode(ecg3, INPUT);
  lcd.begin();                 //Init the LCD
  lcd.backlight();             //Activate backlight
  lcd.home();
  lcd.setCursor(0,0);lcd.print("IOT BASED");
  lcd.setCursor(0,1);lcd.print("VENDILATOR MACHINE");
  delay(2000);
  lcd.clear();
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C
port, 400kHz speed
  {
    Serial.println("MAX30105 was not found. Please check wiring/power.
");
    while (1);
  }
  Serial.println("Place your index finger on the sensor with steady
pressure.");
```

```
  particleSensor.setup(); //Configure sensor with default settings
particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to
indicate sensor is running
  particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
}

void loop() {

long irValue = particleSensor.getIR();

  if (checkForBeat(irValue) == true)
  {
    //We sensed a beat!
    long delta = millis() - lastBeat;
    lastBeat = millis();

    beatsPerMinute = 60 / (delta / 1000.0);

    if (beatsPerMinute < 255 && beatsPerMinute > 20)
    {
      rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the
array
      rateSpot %= RATE_SIZE; //Wrap variable

      //Take average of readings
      beatAvg = 0;
```

```
}

    for (byte x = 0 ; x < RATE_SIZE ; x++)
      beatAvg += rates[x];
    beatAvg /= RATE_SIZE;
  }
 }
 Serial.print("irvalue  :");
 Serial.println(irValue);
 oxygenval=map(irValue,75000,90000,94,100);
 if(oxygenval<=94){oxygenval=0;}
 if(oxygenval>=100){oxygenval=100;}

 Serial.print("OXygen :");
 Serial.println(oxygenval);

  t = dht.readTemperature();
   lcd.setCursor(0,0);lcd.print("T:");
   if(t < 9){lcd.print("0");lcd.print(t);}
   else if(t < 99){lcd.print(t);}

   h = dht.readHumidity();
   lcd.setCursor(5,0);lcd.print("H:");
   if(h < 9){lcd.print("0");lcd.print(h);}
   else if(h < 99){lcd.print(h);}

 if((digitalRead(ecg2) == 1)||(digitalRead(ecg3) == 1)){
Serial.println('!');
}
```

```
else{
// send the value of analog input 0:
Serial.println(analogRead(ecgv));
ecg=analogRead(ecgv);
ecg = map(ecg,0,700,60,120);
}
    lcd.setCursor(10,0);
    lcd.print("E:");
    if(ecg>=999){ecg=0;}
    if(ecg  <=  9){lcd.print("00");lcd.print(ecg);}
    else if(ecg <= 99){lcd.print("0");lcd.print(ecg);}
    else if(ecg <= 999){lcd.print(ecg);}
  sendata();

  delay(500);
}
void sendata()

{
  Serial.print('*');
 if(t <=  9){Serial.print("00");Serial.print(t);}
 else if(t <= 99){Serial.print("0");Serial.print(t);}

 else if(t <= 999){Serial.print(t);}
 //humidity value
 if(h <=  9){Serial.print("00");Serial.print(h);}
 else if(h <= 99){Serial.print("0");Serial.print(h);}
```

```
else if(h <= 999){Serial.print(h);}
if(ecg <= 9){Serial.print("00");Serial.print(ecg);}
  else if(ecg <= 99){Serial.print("0");Serial.print(ecg); }
  else if(ecg <= 999){Serial.print(ecg);}

 if(bpm1 <= 9){Serial.print("00");Serial.print(bpm1);}
  else if(bpm1 <= 99){Serial.print("0");Serial.print(bpm1); }
  else if(bpm1 <= 999){Serial.print(bpm1);}

  if(spo <= 9){Serial.print("00");Serial.print(spo);}
  else if(spo <= 99){Serial.print("0");Serial.print(spo); }
  else if(spo <= 999){Serial.print(spo);}
```