VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590014

PROJECT REPORT

ON

Implementation of a Prototype Automatic Compression Mechanism for Bag Valve Mask

Submitted in Partial fulfilment of the Requirements for 8th Semester

Bachelor of Engineering in

Electronics and Communication Engineering

by

CHIRAG S	**1KG19EC020**
DHANUSH K	**1KG19EC024**
SAMARTH SRINIVAS	**1KG19EC085**
DHEVA C	**1KG20EC400**

Under the Guidance of

Mr. GOPALA KRISHNA MURTHY C.R

Asst. Professor,

Department of ECE, KSSEM

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

K.S. SCHOOL OF ENGINEERING AND MANAGEMENT BENGALURU-560109

2022-23

K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **Implementation of a Prototype Automatic Compression Mechanism for Bag Valve Mask** is carried out by Chirag S (1KG19EC020), Dhanush K (1KG19EC024), Samarth Srinivas (1KG19EC085) and Dheva C (1KG20EC400), bonafide students of KSSEM in partial fulfillment for the award of Bachelor of Engineering in Electronics and Communication Engineering under Visvesvaraya Technological University, Belgavi during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Project report and deposited in the department library. The report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Internal Guide	Head of the Department	Principal
Gopala Krishna Murthy C R Assistant Professor Dept. of ECE	Dr. K Senthil Babu Professor & Head Dept. of ECE	Dr. K Rama Narasimha Principal & Director KSSEM
	Professor & Head Dept. of Electronics & Communication Engineering K. S. School of Engineering & Management Bangalore-560 109	Dr. K. RAMA NARASIMHA Principal/Director K S School of Engineering and Management Bengaluru - 560 109

Name of the Examiners **Signature with Date**

1

2

ACKNOWLEDGEMENT

The successful completion of this Project was made possible with the help of guidance received from our faculty members. We would like to avail this opportunity to express our sincerethanks and gratitude to all of them.

We are grateful to our management for providing the necessary infrastructure and an ambience environment to work. We express our profound gratitude to **Dr. K Rama Narasimha**, **Principal/Director**, KSSEM, Bengaluru for giving us opportunity and support to complete the project phase-II.

We are also grateful to our management for providing the necessary infrastructure and an ambience environment to work. We express our profound gratitude to **Dr. K Senthil Babu**, **Professor & Head**, Department of Electronics & Communication Engineering, KSSEM, Bengaluru for giving us opportunity and support to complete the project phase-II.

We are grateful to our guide **Mr. Gopalkrishna Murthy C R**, **Assistant Professor**, Department of Electronics and Communication Engineering, KSSEM, Bengaluru for his valuablesuggestions and advice throughout our work period.

We would also like to thank all the staff members of Department of Electronics and Communication Engineering for their support and encouragement. Finally, we would like to thank all of our friends without whose help and encouragement this project would have been impossible. Definitely most, we want to thank our family. Words cannot express our thanks to our family members for all their love and encouragement.

CHIRAG S	**1KG19EC020**
DHANUSH K	**1KG19EC024**
SAMARTH SRINIVAS	**1KG19EC085**
DHEVA C	**1KG20EC400**

TABLE OF CONTENTS

CHAPTERS	PAGE NO
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 LITERATURE SURVEY	4
CHAPTER 3 POPOSED METHODOLOGY	16
3.1 BLOCK DIAGRAM	17
3.2 FLOW CHART	19
CHAPTER 4 IMPLEMENTATION	21
4.1HARDWARE & SOFTWARE REQUIREMENTS	22
CHAPTER 5 TESTING	44
HAPER 6 RESULTS & DISCUSSIONS	53
CONCLUSIONS & FUTURE SCOPE	57
REFERENCES	59

LIST OF FIGURES

Figure.3.1 Block Diagram 17

Figure.3.2 Automatic Control 19

Figure.3.3 Manual Control 21

Figure.4.1 Arduino Uno R3 24

Figure.4.2a Arduino Nano Front 26

Figure.4.2b Arduino Nano Rear 26

Figure. 4.3 Max30100 Oxygen Level Sensor 29

Figure.4.4 SIM900A GSM module 30

Figure.4.4a SIM900A GSM module pinout 31

Figure.4.5a TB6600 Stepper Motor Driver 33

Figure.4.5b TB6600 Stepper Motor Driver pinout 34

Figure.4.6 NEMA 17 Stepper Motor pinout 36

Figure.4.7 20x4 LCD Display 38

Figure.4.8 Bag Valve Mask 40

Figure.4.9 Flow Sensor 41

Figure.4.10 Arduino IDE 42

Figure.5.1a Spo2 Working 45

Figure.5.1b Oxygen Value 46

Figure.5.1c Spo2 Circuit Diagram 47

Figure.5.2a Stepper motor Working 48

Figure.5.2b Stepper Circuit Diagram 48

Figure.5.3a GSM Working 49

Figure.5.3b GSM Circuit Diagram 50

Figure.5.4a Flow Sensor with BVM 50

Figure.5.4b Flow Sensor Circuit Diagram 51

Figure.5.4c Initial Values of Flow Sensor 51

Figure.5.4d Final Values of Flow Sensor 51

Figure.5.5a Initial Stage BVM Bag with stepper 53

Figure.5.5b Final Stage BVM Bag with stepper 53

Figure.6.1 Prototype Front View 55

Figure.6.2 Prototype Inside View 1 55

Figure.6.3 Prototype Inside View 2 55

Figure.6.4 Spo2 Sensor 56
Figure.6.5 Automatic Mode (Normal Condition) 57
Figure.6.6 Automatic Mode (Abnormal Condition) 58
Figure.6.7 Manual Mode 59

CHAPTER 1

INTRODUCTION

The last few months have seen an increased demand for ventilators in the treatment of patients with COVID-19, a fact that led to a ventilator shortage worldwide. The consequence of this shortage is calamitous, especially in deprived areas. Even well-equipped hospitals have developed protocols forsharing the same ventilator between two patient a dubious practice because it opens up the possibilityof not only sharing bacterial and viral load among patients, but also provoking untoward harm. As anattempt to face the worldwide problem of ventilator shortage, researchers have started an initiative ofproducing low-cost, open-source ventilators.

Researchers agree that mechanical ventilation can harm lungs, provoking a condition known as ventilation-induced lung injury The two most common types of damage are volutrauma and atelectrauma. Volutrauma appears when the ventilation in excess distends the airways andalveoli, causing over-stretching of the corresponding lung parenchyma. Volutrauma causesan inflammatory reaction, which eventually leads to the rupture of the alveolar walls and edema. Atelectrauma, in contrast, seems to be caused by insufficient ventilation; poor ventilation allows alveolar units to collapse and reopen, in a repetitive, sequential movement, which may lead to injurious well. To avoidatelectrauma, most researchers recommend clinicians doing mechanical ventilation with positive end-expiratory pressure (PEEP). PEEP has become a ubiquitous tool to prevent atelectrauma, gaining evidence after the seminal results in However, using PEEP to reduce lung injury is debatable since recent findings suggest that PEEP causes other harm like lung inflammation and edema formation. These investigations together indicate that the full understanding of the mechanical ventilation processrequires more research mechanical ventilators undoubtedly help patients suffering from acute respiratory problems.

Since mechanical ventilators potentially expose the patient’s lungs to damage, all initiatives of constructing low-cost mechanical ventilators must provide the regulation of not only the lung’s pressure but also the positive end-expiratory pressure (PEEP)—two points for concern. The firstpointinvolves regulating the machine to prevent excessive pressure, which is a side effect of thesurplus ofenergy from the ventilator machinery. We developed a novel method that monitorsthe patient’s pulmonary condition to mitigate the chance of occurring those undesired spikes in the lung’s pressure.
The concern on the ventilator’s design is that of assuring PEEP. A limitation of the mechanical ventilator shown here is that it does not account for PEEP. Yet PEEP valves are commercially

available and can be readily adapted in the inspiratory limb. For this reason, the no-PEEP- monitoring conditiondoes not prevent the use of this low-cost ventilator in clinical trials.

This project's main contribution is to describe the construction of a low-cost, open-source mechanical ventilator for patients with COVID-19. Recall that technology can be considered as low cost if the corresponding deployment can be produced by using inexpensive spare parts. Anelectronic device that has been largely used in these low-cost initiatives is the Arduino, a versatile small-size single-board computer. The Arduino is a critique tool in this paper since it acts as the "brain" of the machinery, as detailed in the sequel.

In summary, this project presents two main finding: (i) construction of a low-cost, open-source mechanical ventilator for patients with COVID-19, and (ii) deployment of a method that monitors thepressure in the patient's lungs and acts on the machinery accordingly.

HISTORY

Early Days:

In the late 18th century, Scottish physician John Dalrymple described using a bellows to ventilate a patient who was drowning. This method was later refined by French physician Eugene Bouchut in 1876, who developed a hand-operated rubber bulb, called the insufflator, to force air into the lungs of patients suffering from respiratory failure. However, these methods were primitive and required manual intervention.

Iron Lung:

The iron lung, also known as the negative pressure ventilator, was invented in the early 20th century and was first used to treat polio patients during the polio epidemics of the 1930s and 1940s. The iron lung worked by creating negative pressure around the patient's body, which caused the chest to expand and draw air into the lungs. Patients were placed inside the iron lung, with only their head sticking out, and the machine would mimic the natural breathing process. The iron lung was large and cumbersome, and patients had to remain inside the machine for long periods.

Positive Pressure Ventilators:

Positive pressure ventilators were first introduced in the 1950s. These machines delivered air into the lungs under positive pressure, rather than creating negative pressure around the patient. The first positive pressure ventilators were large and bulky, and required a team of healthcare professionals to operate. They were typically used in operating rooms and ICUs.

In the 1960s, the first portable positive pressure ventilator was developed, which allowed patients to receive ventilation outside of the ICU. This was a major breakthrough, as it allowed patients to be transported to other areas of the hospital without having to disconnect from the ventilator. Over the years, positive pressure ventilators have become more compact and portable, making them more accessible to patients.

Modern Ventilators:

Today, ventilators are highly sophisticated machines that can be programmed to deliver specific amounts of air and oxygen to the patient, depending on their needs. Modern ventilators are equipped with a range of features, including alarms and safety measures to ensure that patients receive the appropriate amount of air and oxygen.

In recent years, there have been major advances in the technology used in ventilators. For example, some ventilators now come equipped with pressure sensors that can detect changes in the patient's lung function and adjust the ventilation accordingly. Other advances include the use of non-invasive ventilation, such as CPAP (continuous positive airway pressure), which can help patients breathe without the need for intubation.

The COVID-19 pandemic has highlighted the critical role that ventilators play in modern medicine. The virus can cause severe respiratory distress in some patients, and ventilators have been used to help these patients breathe while their bodies fight off the infection. In response to the pandemic, companies around the world have been ramping up production of ventilators to meet the increased demand.

CHAPTER 2

LITERATURE SURVEY

SYNCHRONIZED INTERMITTENT MANDATORY VENTILATION MODE CONTROL USING PULSEOXIMETER

Published by International Conference on Computing, Mathematics and Engineering Technologies in the year 2008. This paper is based on the Spo2 sensor which can determines the oxygen level by light Emitting Diodes and light detector. The paper presenter has used the oximeter and it is configured with the Raspberry Pi to maintain the oxygen level in our body. Outof which, Synchronized Intermittent Mandatory Ventilation (SIMV) is an important and frequently used mode. This mode is controlled with knob dial provided on customary ventilators after adjusting pre requiredsettings according to physician prescription. Dialing up and down the knob requires a trained operator which not only involves human interference that may cause a human error but also time delay. These risking factors are life-threatening and unavoidable as breathing is vital for life. In this work, we propose Synchronized Intermittent Mandatory Ventilation mode control using Pulse Oximeter that is cheap, accurate and easy to use. It checks blood saturation level with the help of Pulse Oximeter sensor placed on a finger or earlobe, provide this signal to the main architectural unit which rotates knob as per requirement of oxygen. To measure this vital sign and to control the amount of oxygen-deliveringto the patient, an instrument is needed that will provide continuous and immediate available data for themonitoring individuals oxygen saturation level.[1]

A NOVEL METHODOLOGY FOR THE DESIGN OF A PORTABLE VENTILATOR.

Published by International Journal of Innovative Technology and exploring engineering(IJITEE) in the year 2019. This paper deals with hardware design of the lab model of ventilator which they have used Arduino has the controller and they pressure sensor to measure the air flow in outlet ofthe bag. In this project they made a prototype device and they have used the needle valve and a potentiometer to increase the pumping rate and they have displayed the readings of the sensor. Ventilators play a vital role in human’s life. It is a pieceof equipment, software program or product system that is used to increase, maintain, orimprove the functional capabilities of persons with disabilities in breathing period. This paper deals with the hardware design of a lab model ventilator. Aprototype model of the ventilator hasbeen designed and tested. A prototype device to assist the patientswho 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.[2]

MECHANICAL VENTILATOR CONTROL SYSTEM USING LOW-COST PRESSURE SENSORS.

Published by IEEE in the year 2021. In this project they have implemented the concept of comparing two air pressure sensor in mechanical ventilator. At the bottom of the paper they have usedtwo pressure sensor to measure the air flow at output which was one at bag outlet and another was in mask by comparing the input and output they have maintained the consistence of air flowing in the mask. . The system tries to maintain the PEEP value of 5 cmH2O, and the feedback obtained ranges from 2.7-5.16 cmH2O. At the same time, the expected flowrate value of 55 L/min can be maintained at a value of 53.9 - 59.5 L/min. The tidal volume, which functions as a limiter for inspiration and expiration, is set at a value of 400ml; the feedback given by the sensor varies between 416 ml - 436 ml. Nevertheless, on the other hand, this system needs to be developed further because there are problems with sensor precision. According to the experimental results, the problem is influenced by several things that can be significant in system reliability. First, the pressure sensor selection is a dominant factor in producing poor feedback to the microcontroller as the central controller. In addition,it isnecessary to consider using a Mass Air Flow sensor to get better flow rate and volume values. Finally, a PID or fuzzy logic control system will produce more precise Ambu bag suppression infuturesystem development.[3]

DIY VENTILATOR USING ARDUINO WITH BLOOD OXYGEN SENSING FOR COVID PANDEMIC.

Published by International Research journal of engineering and technology (IRJET) In the month of July 2021. This paper have the three use full functions like tidal volume, bpm and IE ratio. The user or doctor can have the presets as they required for the patient. The DIY ventilator has 3 functions and they can be changed has per the requirements and it will be displayed in the lcd screen but these is at beginner phases. While breathing heartbeat level identified are low this component can be performed. The LED screen is utilized to show the breathing heartbeat levels. Likewise, in a patient basic condition or breathing issue ringer is fitted in the framework to sound a ready when any irregularities are identified. The ventilator we here plan and foster utilizing Arduino envelops this large number of necessitiesto create a dependable yet cheap DIY ventilator to help in the midst of pandemic. Sure this work will acquire more noteworthy consideration. This is at the beginning phases of planrequired further turns. This paper is an itemized clarification of delivering minimal expense, open source mechanical ventilators for patients.[4]

AN INTRODUCTION TO LOW-COST PORTABLE VENTILATOR DESIGN.

Published by International Conference on Advance Computing and Innovative Technologies in Engineering 2021.This paper Provides all idea of origin of modern mechanical Ventilators. The proposed ventilator is assumed to have better working performance than already available in market in very low cost. The low-cost portable ventilator is introduced with the help of suitable drawing and control loop diagram. The proposed low-cost portable ventilator is an inexpensivealternative to the exorbitant purchasing cost hospital ventilators. In this paper a low-cost portable ventilator designed is introducedwith the help of suitable drawing and control loop diagram. The proposed ventilator can provide 500-600ml tidal with a continuous working ability especially for pneumonia patients. There are three user inputs (bpm, wave volume, and 1: E ratio) set by three potentiometer terminal. Future device times included will be the addition of a LED display to display air pressure level and input settings too battery power status. Air pressure is considered bypressure sensor connected to the sensor output in BVM for ensuring that the patient is not harmed. If the pressure too high the same pressure sensor as the pressure sensor used by the start of helping alarm triggers power which warns the doctor care for the patient. As an alternative to safety prevent excessive exaggeration, future calculations consolidate the pressure life valve of the unit.[5]

A REVIEW OF OPENSOURCE VENTILATORS FOR COVID-19 ANDFUTURE PANDEMICS.

Published by F1000 Research in the year 2020. These articles are analyzed to determine thedesign in spirit as well as practical details. There is clear technical potential for ventilator shortages duringthis & future pandemics using opensource ventilator that can be rapidly fabricated. The results ofthis review found that the tested and peered viewed systems lacked complete documentation and the open systems that were documented were either at the very early stages of design (sometimes withouteven prototype) and were essentially only basically tested (if at all). With the considerably larger motivation of an ongoing pandemic, it isassumed these projects will garner greater attention and resource s to make significance progress to reach a functional and easily-replicated system. Future work is needed to achieve the potential of this approach by developing policies, updating regulations, and securingfunding mechanisms for the development and testing of source ventilators for both the current COVID-19 pandemic as well as for future pandemics and for everyday use in low-resource.[6]

RECENT ADVANCES IN LOW-COST, PORTABLE AUTOMATED RESUSCITATOR SYSTEMS TO FIGHT COVID-19.

Published by Health & technology in the year 2021. This study aims to provide lowcost ventilators which is easy to assemble. Automated resuscitator systems are installed by providing positive pressure to the lungs. The system is low-cost easy to assemble, portable ventilator to fight thePandemic. The exhaust value should be fitted to stop the accumulation of Co2. Various designs havebeen proposed, focusing on meeting basic requirements for artificialventilation to fight the ongoing pandemic. But some people are against the usage of these mechanical ventilators in real-life situations, owing to poor reliability and inability of these designs to meet certain clinical requirements. Each design has its own merits and demerits, which need to be addressed for proper designing. Therefore, 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. By going through existing literature, we have recognized multiple factors infuencingdevice performance and how these factors need to be considered for efcient device operation. Thus, the collective efforts of engineers, doctors, scientists and even policy makers are needed for better designing and implementation of these low-cost portable mechanical ventilators to efectively fght againstthe current and future pandemics.[7]

IOT ENABLED VENTILATOR MONITORING SYSTEM FOR COVID-19 PATIENTS.

Published by The Electrochemical Society Advancing solid State & electrochemicalscience &technology. In the year 2021.This study aims to Provide a monitoring & control systemfor a Ventilator based on IoT. It is used to test the data transmission function with the IoT protocol. This research produces the control & monitoring system for a mechanical ventilator. The motion of the gripper creates pressure & releases pressure on an AMBU Bag. 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 andpressure of the air delivered to thelungs. The rate of pressure exerted is measured as the velocity of air flowing into the lungs. The control and monitoring system uses Arduino components as the main control. The Bluetooth based android application chosen is Virtuino, because it has a user-friendly interface and can work efficiently with Arduino to control and monitor variables on the ventilator via smart devices.[8]

A PORTABLE BVM-BASED EMERGENCY MECHANICAL VENTILATOR.

Published by IEEE 19^{th} World Symposium on Applied Machine Intelligence and Informatics in the year 2021.The aim of the paper is to present developed ventilator based on BVM which is alternative to mechanical ventilator in critical situations like covid-19. The character of IoT is the nextsignificant element of this developed ventilator. The mechanical andcontrol design are presented in the next section. Finally, we experimentally verify developed ventilator with focus on measured pressure of patient airways. The presented results show a potential of developed ventilator to be used at practical level. r. Our requirements to development of low-cost mechanical ventilator were: simplicity, robustness, portability, fast reproducibility, and utilization in hospitals, in field hospitals and in households. The character of IoT is the next significant element of our developed ventilator. Mechanical side of ventilator is designed to be fast reproducible in the case of critical situations, where shortage of professional ventilators would occur. The ventilator has the potential to be used also inhouseholds for patient without pulmonary pathology. The developed mechanical ventilator is designedespecially for the patients with light course of lung disease.[9]

DESIGN & IMPLEMENTATION OF PORTABLE EMERGENCY VENTILATOR FOR COVID-19 PATIENTS.

Published by Advances in Science and Engineering Technology International Conferences. Inthe year 2022. This project is based on Arduino mc to control a stepper motor, a pressure sensor, bodytemperature sensor and a pulse oximeter sensor. It is low cost compared to already existing ventilator.It was designed, implemented, and tested with very good results such as pulse oximeter sensor error was 1.02% and temperature sensor error was 0.28%.

The ventilator is extremely necessary for patients that are in dire need of assistance with their breathing caused by the effects of COVID-19. Hospitals only have a fair number of commercial ventilators because they are expensive, large, immobile, and difficult to move. Thegoal of this paper is to design and build a portable emergency artificial ventilator that is reasonably affordable, small compared to the commercial ventilator, easier to move, could be used at home and can act as a worthy replacement for the commercial ones. The proposed emergencyventilator is based on Arduino microcontroller to control a stepper motor, a pressuresensor, body temperature sensor and a pulse oximeter sensor. This has also been supported by the final implemented prototype of lower cost (with respect to commercial products) and lowerpower consumption.[10]

MODEL-FREE VOLUME AND PRESSURE CYCLED CONTROL OF AUTOMATIC BAG VALVE MASK VENTILATOR.

Published by AIMS Bioengineering in the year 2021. The BVM ventilator comprises a man-made manual breath unit bag and paddles for squeezing the AMBU bag which is popularin medical aid settings. Therefore, a model-free control approach is utilized successfully to style a controller for our BVM ventilator model with a PEEP valve and a HEPA filter. The BVM ventilator is required to travel airflow through the system to the patient's lung with the specified volume for every breath cyclewithin a threshold air pressure. Since the AMBU bag isstraightforward to be deformed over time, it's difficult to get mathematical modelling for constructing a reliable controller. Therefore, a model-free control (MFC) control approach is utilized successfully to style a controller for our BVM ventilator model with a PEEP valve and a HEPA filter. Some experimental scenarios are administered to gaugethe effectiveness of the proposed controller for the BVM ventilator to control the airflow and control air pressure mode. For the following work, the BVM ventilator will be validated on a suitable WET Lab (University of Medicine and Pharmacy at Ho Chi Minh city) with an animal test in the medicalprocess. Experiments on large animals like buffalo, pigs, sheep, and others can be used to evaluate theBVM ventilator.[11]

LOW-COST, OPEN-SOURCE MECHANICAL VENTILATOR WITH PULMONARY MONITORING FOR COVID-19 Patients.

This paper shows the construction of a low-cost, open-source mechanical ventilator. The motivation for constructing this kind of ventilator comes from the worldwide shortage of mechanical ventilators for treating COVID-19 patients—the COVID-19 pandemic has beenstriking hard in some regions, especially the deprived ones. Constructing a low-cost, open- source mechanical ventilator aimsto mitigate the effects of this shortage on those regions. The equipment documented here employs commercial spare parts only. This paper also shows a numerical method for monitoring the patients’ pulmonary condition. The method considerspressure measurements from the inspiratory limb and alerts clinicians in real-time whether the patient is under a healthy or unhealthy situation. Experimentscarried out in the laboratory that had emulated healthy and unhealthy patients illustrate the potential benefits of the derivedmechanical ventilator. Experiments carried out in the laboratory that had emulated healthy and unhealthy patients illustrate the potential benefits of the derived mechanical ventilator.[12]

REVIEW FOR ARDUINO BASED PORTABLE VENTILATOR FOR COVID-19

This paper provides an idea of the origin of modern mechanical ventilators. Theproposed ventilator prototype here is assumed to have better working performance than alreadyavailable around. This ventilator consists of blood oxygen sensor & a sensitive pressure monitoring system all at very low cost. The ventilator in this paper where designed and developed using Arduino encompass all therequirements to develop a reliable yet affordable whilst matching all the conditions & marks of an original mechanical ventilator along with its property of portability. This can still be the early stages of such designs & might need further development eventually gaining more attention. It is a big sourcefor emergency purposes and even for everyday use in low-resource settings. The ventilator we here design and develop using Arduino encompass all the requirements to develop a reliable yet affordable whilst matching all the conditions and marks of an original mechanical ventilator along with its property of portability. This ventilator will help in a situation like COVID-19 when the whole world facing difficulties related to ventilators and thereby can also be used in other critical situations or mass casualties. It is a big source for both the current pandemic situation and emergency purposes. It is a big source for both the current pandemic situation and emergency purposes and even for everyday usein low resource setting.[13]

DYNAMIC CHARACTERISTICS OF A MECHANICAL VENTILATION SYSTEM WITH SPONTANEOUS BREATHING

Mechanical ventilation is an important & effective method for the treatment of pulmonary diseases patients with spontaneous breathing. Spontaneous breathing refers to the physiologicalbreathing activity caused by the respiratory muscle. The pneumatic model of a mechanical ventilation system with spontaneous breathing established in this paper can simulate the treatment plans for patients with respiratory diseases and then calculate and optimize the pressure support level. The dynamic characteristics (mainly flow characteristics, tidal volume characteristics and pressure characteristics) of the mechanical ventilation system are directly influenced by variations in C, A and1Pmus. This pneumatic model can be used in the design and performance optimization of active lungsimulators, ventilators, and some other medical devices. However, asynchrony is a common problem in mechanical ventilation during spontaneous breathing activity. In the next step, we will focus on the identification of asynchrony and novel control solutions for ventilators to solve this problem.[14]

DESIGN AND STUDY OF A PORTABLE HIGH-FREQUENCYVENTILATOR FOR CLINICAL APPLICATIONS:

This paper presents the design of a portable high-frequency ventilator & a study of its practicality for further clinical medical applications. This paper contains developed portable HF-ventilator with reconfigurable $o_2$flow rate, applied pressure & air volume for the needs of individualpatients. The proposed system achieves a tunable DP range of 5-45 psi, an I:E ratio of 10-70% oxygen concentration of 20%-100% & a flow rate of 0-40L/min. allparameters are displayed on LCD monitor & controlled from a computer. The proposed ventilator has a compact size of 20 x 15 x 17 cm3 whichis much smaller than existing HFJVs, and the experimental results suggest the robustness andeffectiveness of the proposed system. With its portable design, configurable features, and instantfeedback control, this HFJV could be employed in clinical as well as non-hospital settings including ambulance and home healthcare, making the medical treatment more convenient.[15]

DESIGN & ANALYSIS OF LOW-COST ELECTRONICALLY CONTROLLED MOBILE VENTILATOR, INCORPORATING mechanized AMBU bag for patients during covid-19 pandemic:

Published by Hindawi journal of healthcare engineering, in the year 30 March 2022. This article has made an attempt to design and simulate an economically available artificial ventilator for the ailing human subjects needing an external respiratory support system in the present pandemic situation in worldwide. The researchers have aimed to design & develop a cost-effective, simple effectively mechanized ventilator unit, which can be assembled without much hassle and can be operated to resuscitate a patient in dire need.Constructing a low-cost, open-source mechanical ventilator aimsto mitigate the effects of this shortage on those regions. The equipment documented here employs commercial spare parts only Although it is simple, less costly instrument cannot exactly replace the ventilators available in the present market at an exorbitant price. Consequently, the inaccuracies induced by thermal and tribological factorsduring the continuous running of the mechanism have not been considered and may be takenas a follow-up study on the proposed design. Also, the algorithm can be improvised toincorporate patient-triggered ventilation byaugmenting sensory units along the airflow circuit that can monitor the actual breathing cycle of the patient and make the AMBU bag actuation complaint to that. [16]

DESIGN AND DEVELOPMENT OF AN AUTOMATED COMPRESSION MECHANISM FOR A BAG- VALVE-MASK - BASED EMERGENCY VENTILATOR:

In response to COVID-19 pandemic, universities and related institutions around the world came up with various mechanical ventilator designs to help cope with the expected shortages of ventilators as the pandemic rages. Many of these designs are based on automating the manual operation of the BagValve Mask (BVM), a ubiquitous resuscitator device used for emergency ventilation or resuscitation of patients with breathing problems. In this paper, the mechanical design and development process for a BVM-based emergency ventilator is discussed. In particular, the evolution of the design from a simple, low-cost device to a more sophisticated system acceptable to pulmonologists and relatedmedical practitioners is documented. In particular, the evolution of the design from a simple, low-costdevice to a more sophisticated system acceptable to pulmonologists and related medical practitioners is documented.[17]

NET-VENT: LOW-COST, RAPIDLY SCALABLE AND IOT-ENABLEDSMART INVASIVE MECHANICAL VENTILATOR WITH ADAPTIVECONTROL TO REDUCE INCIDENCES OF PULMONARY BAROTRAUMA IN SARS-COV-2 PATIENTS:

The recent outbreak of the SARS-CoV-2 pandemic has disrupted healthcare systems allover the world. With the spread of multiple variants, the progress of the pandemic is unpredictable. In critical cases, viral respiratory illness causes fluid build-up in the lungs, requiring the patient to be onventilation to have sufficient oxygen. With an enormous shortage of ventilators worldwide, the creation of an emergency ventilator has become a compelling international engineering challenge. Thispaper demonstrates a distinct proof-of-concept designand working of NeT-Vent: a low cost, rapidly scalable, easy to manufacture and IoT-enabled smart invasive mechanical ventilator which addresses Pulmonary Barotrauma, a Ventilator Induced Lung Injury, whose incidences are being increasingly reported during ventilation of COVID-19 patients and remains unaddressed in most open-source ventilators. Prevention of Pulmonary Barotrauma can attenuate multi-organ failure, thus improving survival in high-risk patients. The proposed ventilator is extensively evaluated on various parameters and comparedwith other open-source ventilators. Project NeT-Vent won the Best Project Award at the University of Queensland Engineering Design hackathon, India 2020 on Ventilator Design.[18]

PROTOTYPING OF A COST EFFECTIVE AND PORTABLE VENTILATOR:

Scripture under sight is defining the robustness and functionalities of ventilator which is not only easily transferable as well as it is very low cost and economics friendly. It is designed under the basic idea of being incorporated in huge human catastrophes in poorly resources enriched environments. Ventilator under the proposed design was being developed with wooden pieces with a weight of6 kg and has a volume of14 x 7 x 9 inches. It functions without human operator as it deliversbreaths through the compression of an orthodox bag-valve mask. Itsatisfies its energy needs from an electric motor having battery power of 12 volts DC. Different functions need to be performed for the purpose of ventilation i.e. pressure and required number of breathes per minute is managed by an easy to use input board comprising of buttons. In addition to that it also contains an alarm of low battery indication system as well as an assist control. This proposed design of ventilator is made up of a cost of $150, but on a massive production it will result into cost benefits and it will be available at a price of $100. This piece of prototype is cost effective as well as energy efficient as far as the present technology in ventilators is concerned. It can be declared as a viable option based upon the above characteristics. Work on various areas of resulting outcomes is also proposed including methodsto confine the deaths to its lowest rate, lessor weight and extended life of the battery. Further testing willbe done for more effective usage of alarms. It will be used for indication of loss of power, loss of breathing circuit integrity and low respiration rate. On the Final stage, tests will be enforced for the sake of testing of ventilator on a lung model to come up the standards set for ventilator to make the product able to compete with the market.[19]

ARDUINO BASED LOW-COST VENTILATOR FOR COVID-19.

This paper provides an idea of working and methodology of building low-cost ventilator. The ventilator prototype uses IOT technology to provide real-time tracking oftemperature and Breaths PerMinutes of the patient. This ventilator model is assumed in marketat a very competitive price. The project highlights the possibility of building a portable, low- cost ventilator that will have the basic functions of a ventilator. This ventilator can be used in hospitals, homes and even in small clinics for a short period of time This project highlights the possibility to build a portable, low cost ventilator that will have all the basic functions of a ventilator This ventilator can be used in hospitals and in homes for a short span of time until professional help is provided.[20]

CHAPTER 3
PROPOSED METHODOLOGY

3.1 BLOCK DIAGRAM

Prototype is briefly explained using block diagram which gives the basic flow of the working of the system.

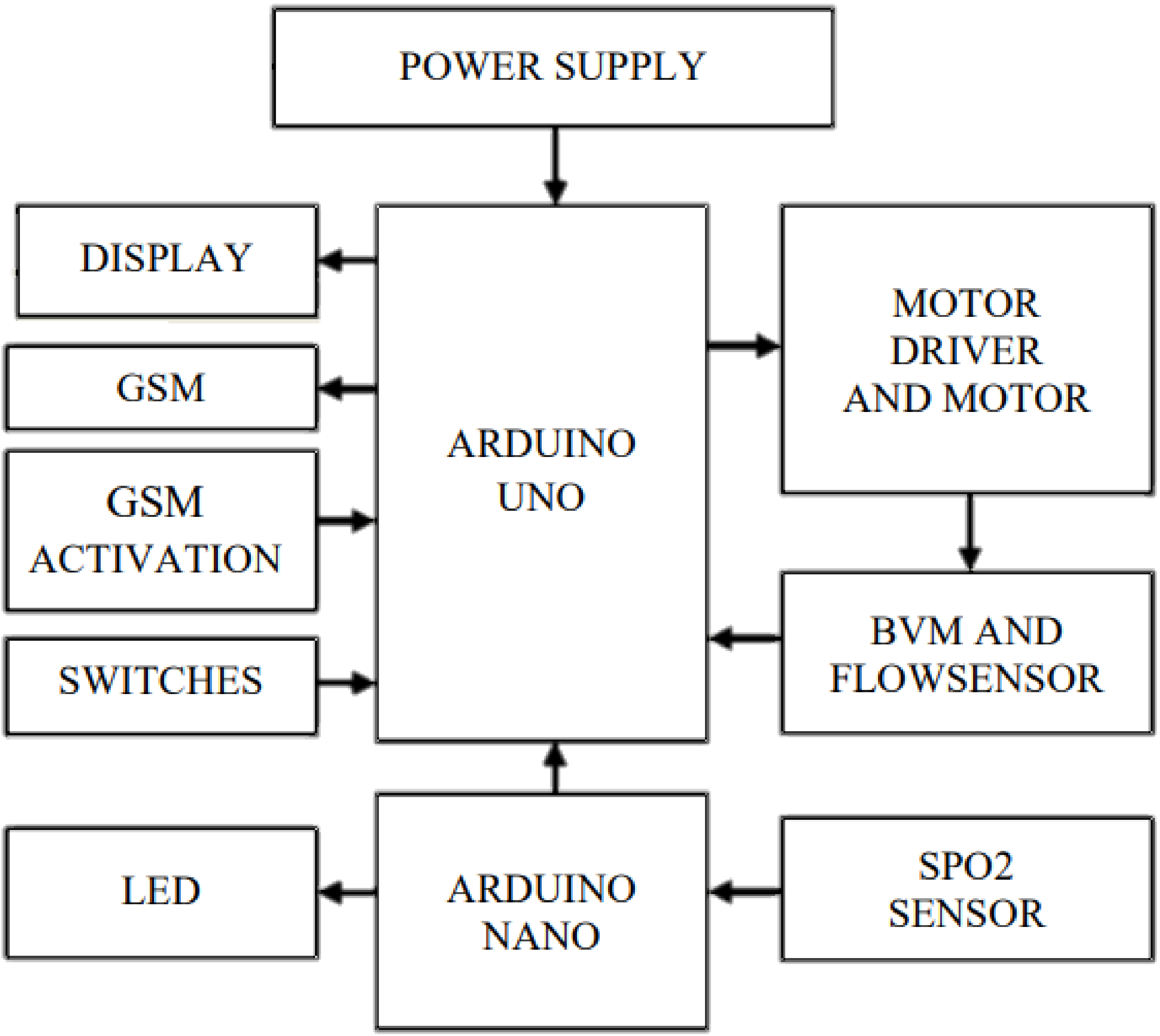


Fig 3.1 Block Diagram

The above figure 3.1 represents the Block Diagram of this project, it has 2 microcontrollers oneis Arduino Uno and other one is Arduino Nano.

Both the Arduino have same digital pins and but the Arduino nano has 8 analog pins Arduino uno is

Powered by 12V the nano can powered by the un controller has it provides gnd, 5v, 3.3v and vin supply. The spo2 sensor MAX30100 which is interfaced with the nano to the pin number the spo2 sensor will measures the oxygen level of the patient and compares the oxygen level if the oxygen level above 85% then the it will indicated in green led indication and the oxygen level in normal, if the oxygen level is below 85% then it indicated in red led indication which means abnormal condition.

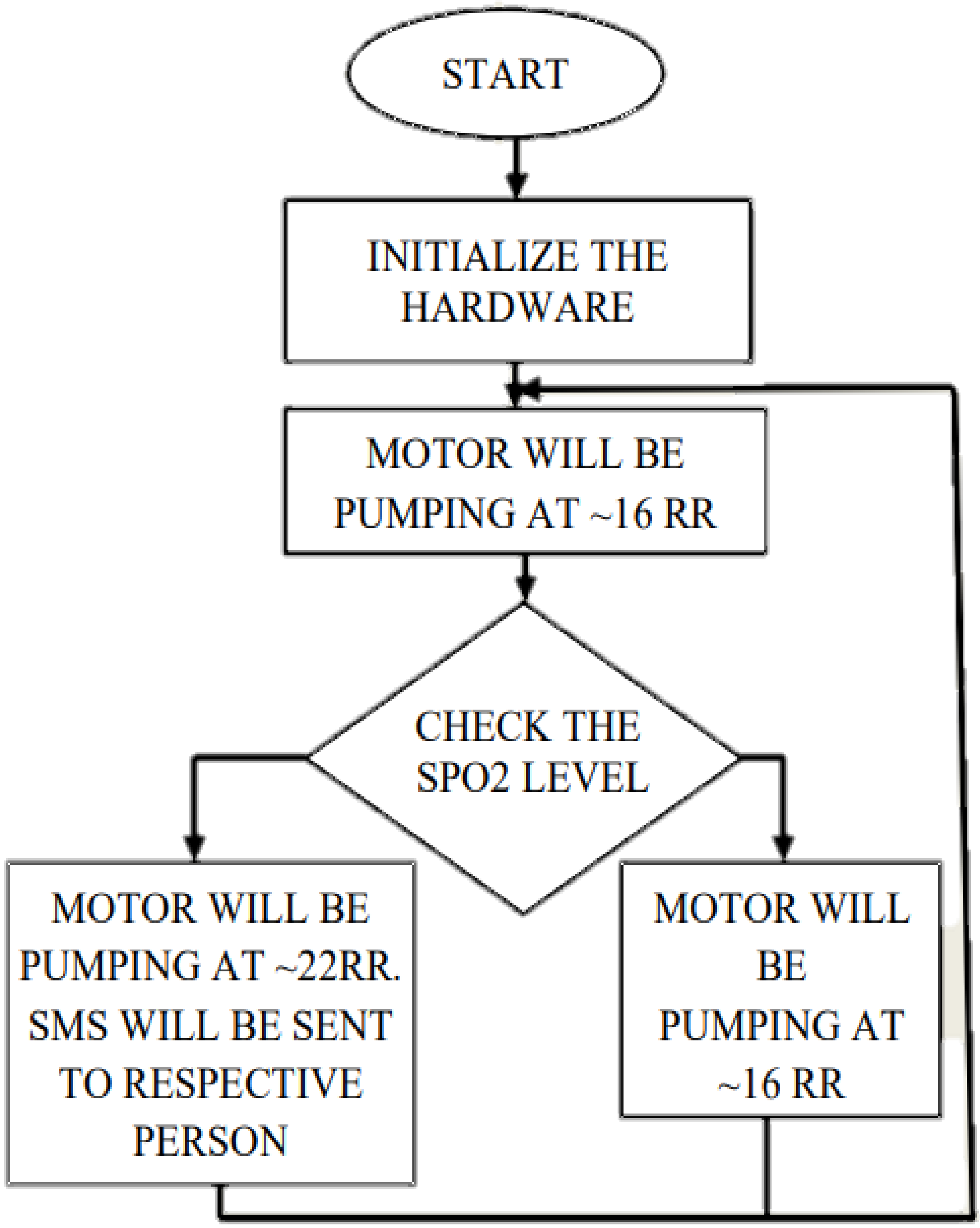
The Arduino nano device is connected to the Arduino uno and Tb6600 motor driver, which is then connected to the Arduino Uno. This motor driver is linked to the Arduino Uno at pins 4 and 5. This allows the motor to be driven in a clockwise or anti-clockwise direction and the step size can be set by the step pin. Then, an Arduino signal activates the motor. This motor is a Nema 17 motor, which is capable of handling a load of 20kg. When the signal is received, the motor starts rotating at around 16 times per minute, if the oxygen level is normal. On the other hand, when abnormal levels are detected, the speed is increased to 22 times per minute.

We have implemented a system using a GSM module that will alert a person if their oxygen level goes below 85%. This system will send a message to the respective person containing the time. After the message is received, the person can check the oxygen level immediately and take the necessary measures to address the issue. The GSM module will also keep track of the oxygen level and alert the person in advanced if the level continues to drop. This way, early detection and prevention of any serious health condition can be achieved. This method is very effective in recognizing any abnormal changes in oxygen saturation levels and taking prompt action accordingly. The timely message sent from the GSM module will allow the person to move quickly and take any necessary medical attention that may be required.

Using bag valve mask (BVM) via Rack & pinion the oxygen will be supplied through the motor rotation mechanism and it is sufficient to breath for a normal person as the studies shows from the literature survey that normal person capacity of breathing is 500ml and our system is capable of supplying ~500ml of oxygen, Flow sensor is used to detect and supply the amount of oxygen required for the patients.

Two switches are given to control automatic and manual toggling and the working will be explained in the flow chart in 3.2, 3.3, and the readings of the oxygen level, flow rate modes and gsm feedback will be displayed in 20x4 lcd display.

3.2 FLOW CHART



3.2 Automatic Control

This project has 2 controlling unit so it has 2 flowcharts, above figure 3.2 represents theautomatic control. So here when the system is on it will initialize the components and displays the current oxygen level and flow rate of the patient. Then spo2 sensor will detect the oxygen level if it is less than 85 bpm (beats per minute) then the RR (Respiratory rate) 22, or else 16.

So, SMS is sent when it is less than 85bpm. The user will receive a sms alert to increase the oxygen level and flow rate. After that the user will turn on the oxygen pump to increase the flow rate and oxygen level. The control unit will monitor the oxygen level and flow rate of the patient and display it on the LCD. If the oxygen level is still less than 85 bpm, then the RR is increased to 22, else it is decreased to 16. After that, the automatic control unit will adjust the oxygen level and flow rate according to the patient's condition. The oxygen level and flow rate are continuously monitored by the control system and displayed on the LCD. The control system will shut down the system if the oxygen level and flow rate are not in the desired range. This project is useful in providing the required oxygen levels and flow rate to patient in a very efficient and safe manner.

The PI (proportional integral) controller checks the overall difference between the measured and reference oxygen level, then based on the setpoint of RR it will activate the LED and send message to the doctors and nurse through the GSM module and day & night mode switching for prompt medical services. As the breathing rate changes, the system auto adjusts the flow rate of the gas to maintain desired concentration of oxygen in the patient. When the oxygen level reaches 85 bpm, then the system again initiates its cycle. The second flowchart is referred as manual control in which temperature, flow rate and O2 saturation levels are monitored manually and accordingly to check the user’s condition. The patient's comfort is monitored using the BPC adaptor and the O2 sensor. The manual control indicates the severity of patient’s condition and it provides a better diagnose for them. All the data and information is displayed in the monitor, so it is convenient for the user. The main advantage of this project is to monitor the oxygen saturation level of patient and also to monitor their comfort. This project also helps in reducing hospital stay time as the patient can be monitored from home. The main advantage of having two flowcharts is that it allows shortening of patient’s hospital stay time and please the patient, while reducing medical cost. By using this project, medical monitoring will be more reliable and accurate with the help of the automated control, hence it helps the medical staff in providing optimal treatment.

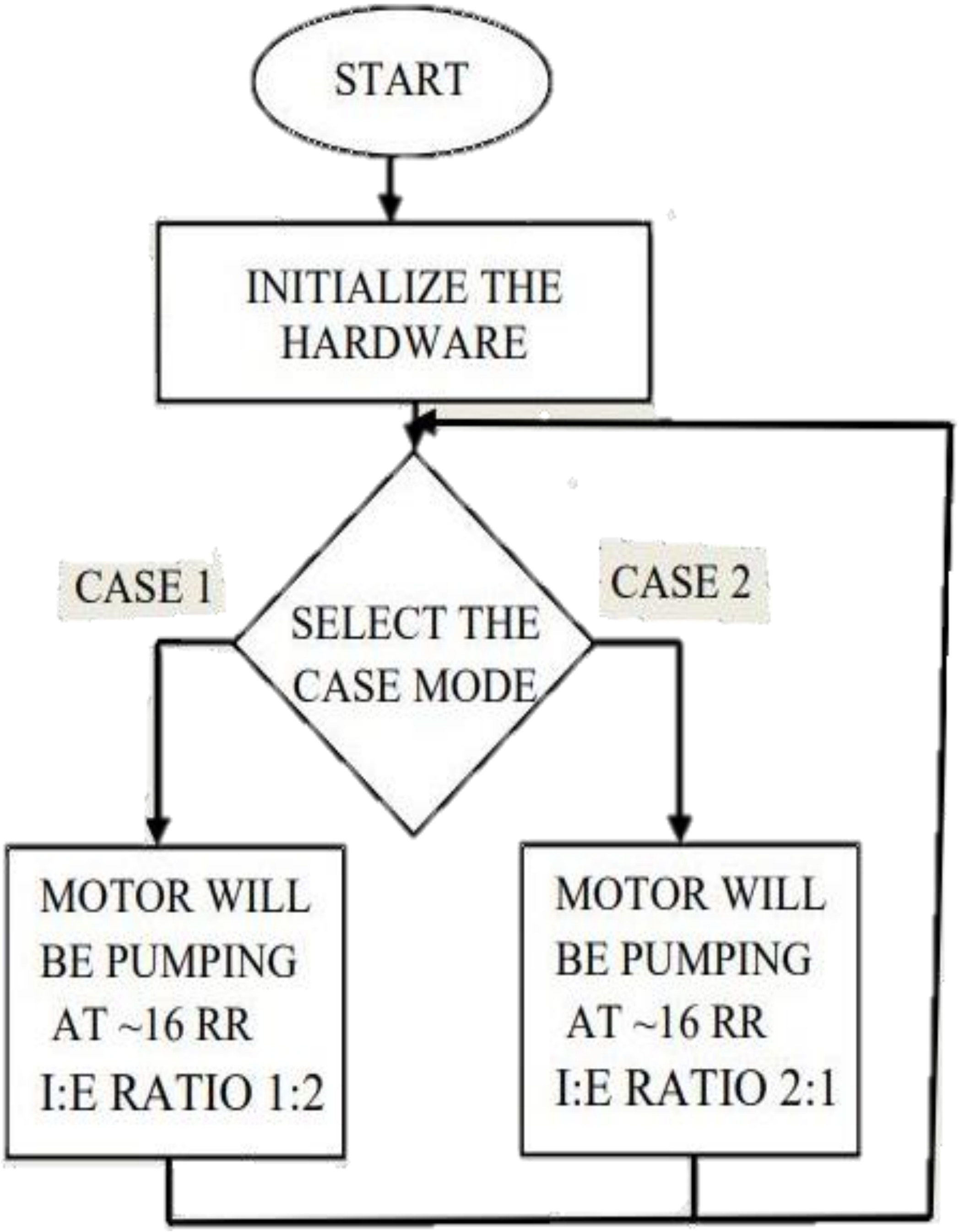


Fig 3.3 Manual Control

The above figure 3.3 represents the manual control, when the toggle is done the control is shifted to manual control. Initialization is done and mode is selected if case 2 is selected motor will pump at 16RR so I:E (inhale to exhale) is 2:1 or viseversa, both are having loops as it is continuous.

The figure 3.3 shows manual control. When the toggle is switched, the control automatically shifts to manual control. As part of the process, initialization is done and a mode must be selected. If case 2 is selected, then the motor will pump at 16RR, which means that the I:E ratio (inhale to exhale) is 2:1 or vice versa. Both the I:E ratio and the 16RR rate have loops, making the process continuous. By having manual control, it helps to ensure that the I:E ratio and the rate of the motor can be adjusted accordingly. This helps to make sure that the process is efficient and effective.

The above figure 3.3 represents the manual control, where the user can manually control a process by changing or toggling from the auto to the manual mode. When this toggle is activated, the sequential control shifts to manual. Initialization is done and mode selection will determine the number of revolutions per minute (RR) and inhale to exhale (I:E) ratio of the motor pump. If case 2 is selected, the motor will run at 16RR and the I:E ration is 2:1. As this is a continuous process, both the manual and the auto mode have looping stages. This manual control would be especially useful in medical conditions that need precise and reliable control.

The Toggling is done using the switches and led is used to represent the normal and abnormal conditions. To select the cases calibration is done and optimized.

CHAPTER 4
HARDWARE & SOFTWARE REQUIREMENTS

4.1 ARDUINO UNO R3

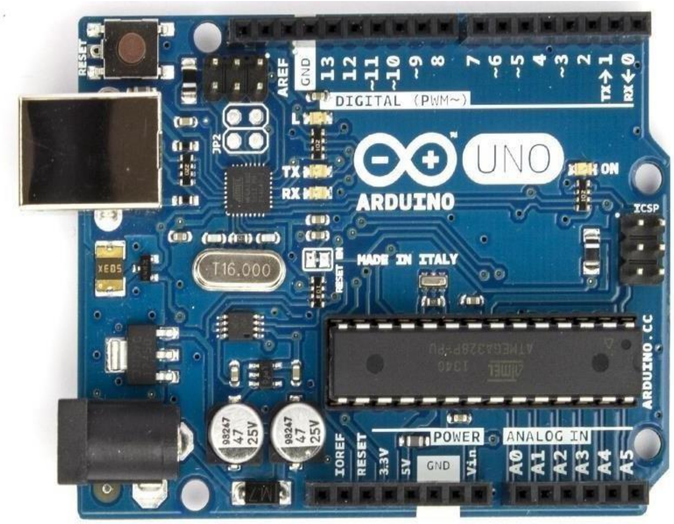


Fig 4.1 Arduino Uno R3

The Arduino Uno Fig 4.1 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 witha USB cable or power it with a AC-to DC adapter or battery to get started.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, anotherArduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port tosoftware on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation fordetails. For SPI communication, use the SPI library.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. Thepower source is selected automatically. External (non-USB) power can come either from an AC- to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center- positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

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

5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V),or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is50 mA.

GND. Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAMand 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

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

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() functionfor details.

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

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

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

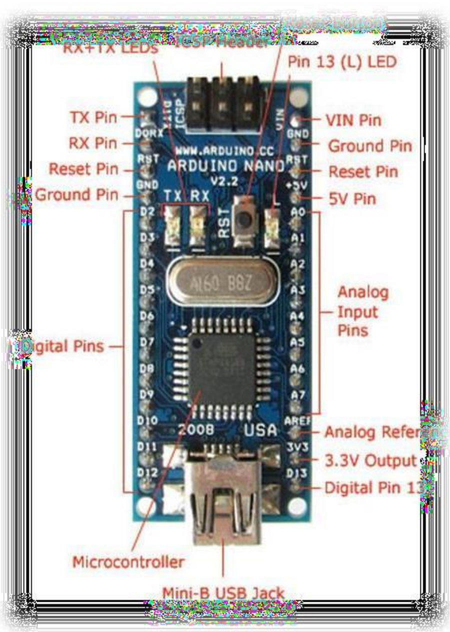
The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board:

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

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

4.2 ARDUINO NANO



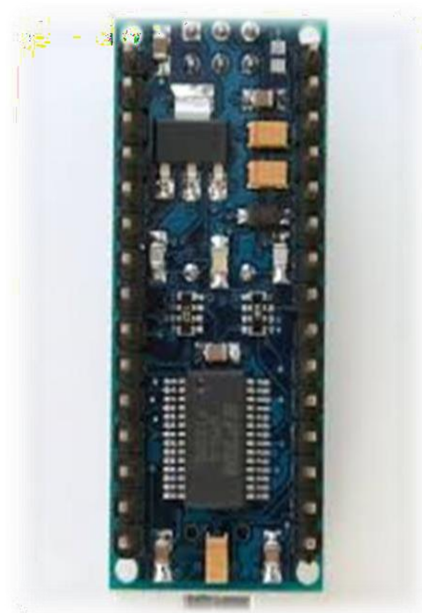


Fig 4.2.a Arduino Nano Front **Fig 4.2.b** Arduino Nano Rear

The Arduino Nano Front Fig 4.2.a and Fig 4.2.b is a Nano rear is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino

Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.

Summary:

4.3 Microcontroller Atmel ATmega168 or ATmega328

4.4 Operating Voltage (logic level) 5 V

4.5 Input Voltage (recommended) 7-12 V

4.6 Input Voltage (limits) 6-20 V

4.7 Digital I/O Pins 14 (of which 6 provide PWM output)

4.8 Analog Input Pins 8

4.9 DC Current per I/O Pin 40 mA.

4.10 flash Memory 16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used bybootloader SRAM 1 KB (ATmega168) or 2 KB (ATmega328)

4.11 EEPROM 512 bytes (ATmega168) or 1 KB (ATmega328) Clock Speed 16 MHzDimensions 0.73" x 1.70"

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V)serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RLon the board channels this serial communication over USB and the FTDI drivers (included withthe Arduino software) provide a virtual com port to software on the computer. The Arduinosoftware includes a serial monitor which allows simple textual data to be sent to and from theArduino board. The RX and TX LEDs on the board will flash when data is being transmitted viathe FTDI chip and USB connection to the computer (but not for serial communication on pins 0and 1). A SoftwareSerial library allows for serial communication on any of the Nano's digitalpins. The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. TheArduino software includes a Wire library to simplify use of the I2C bus; see the documentationfor details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet. **Power** :

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written withthe EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

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

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

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

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

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change theupper end of their range using the analogReference() function. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

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

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

4.3 SPO2 SENSOR:

The MAX30100 shown in Fig 4.3 is an integrated pulse oximetry and heartratemonitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software withnegligible standby current, permitting the power supply to remain connected at all times.

Fig 4.3 Max30100 Oxygen Level Sensor

Benefits and Features

1. Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design

a. Integrated LEDs, Photo Sensor, and High-Performance Analog Front -End

b. Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in-Package

2. Ultra-Low-Power Operation Increases Battery Life for Wearable Devices

a. Programmable Sample Rate and LED Current for Power Savings

b. Ultra-Low Shutdown Current (0.7µA, typ)

3. Advanced Functionality Improves Measurement Performance

a. High SNR Provides Robust Motion Artifact Resilience

b. Integrated Ambient Light Cancellation c.High Sample Rate Capability

d.Fast Data Output Capability

4. **Applications**

5. Wearable Devices

6. Fitness Assistant Devices

7. Medical Monitoring Devices

4.4 GSM

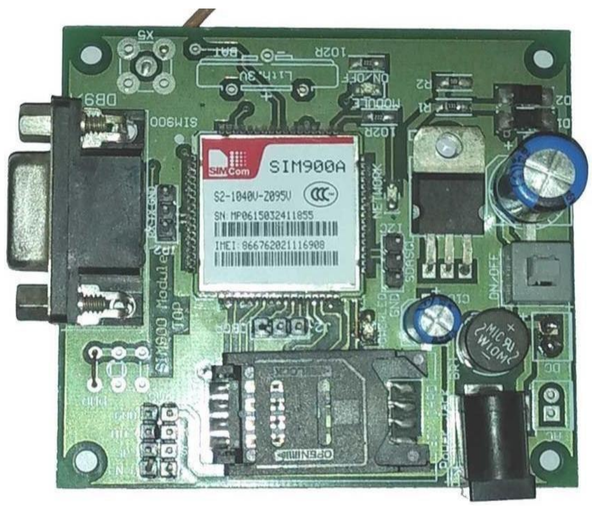


Fig 4.4 SIM900A GSM Module

The SIM900A Fig 4.4 is a readily available GSM/GPRS module used in many mobile phones and PDA. The module can also be used for developing IOT (Internet of Things) and Embedded Applications. SIM900A is a dual-band GSM/GPRS engine that works on frequencies EGSM 900MHz and DCS 1800MHz. SIM900A features GPRS multi-slot class 10/ class 8 (optional)and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

Features and Specifications

Single supply voltage: 3.4V – 4.5V

Power saving mode: Typical power consumption in SLEEP mode is 1.5mA

Frequency bands: SIM900A Dual-band: EGSM900, DCS1800. The SIM900A can searchthe two frequency bands automatically. The frequency bands also can be set by AT command.

GSM class: Small MS

GPRS connectivity: GPRS multi-slot class 10 (default) , GPRS multi-slot class 8 (option)

Transmitting power: Class 4 (2W) at EGSM 900, Class 1 (1W) at DCS 1800

Operating Temperature: -30ºC to +80ºC

Storage Temperature: -5ºC to +90ºC

DATA GPRS: download transfer max is 85.6KBps, Upload transfer max 42.8KBps

Supports CSD, USSD, SMS, FAX

Supports MIC and Audio Input

Speaker Input

Features keypad interface

Features display interface

Features Real Time Clock

Supports UART interface

Supports single SIM card

Firmware upgrade by debug port

Communication by using AT commands

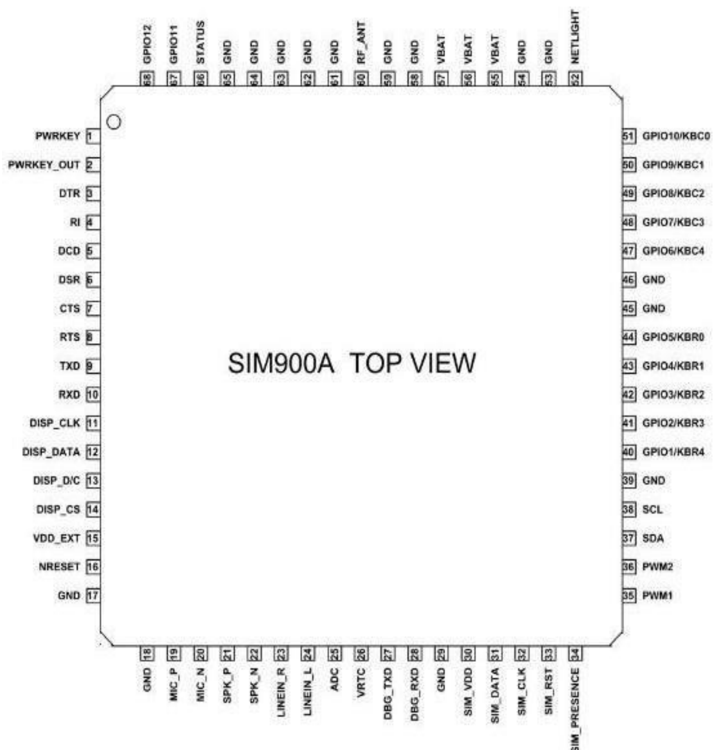


Fig 4.4.a SIM900A GSM Module Pinout

As Shown above Fig 4.4.a, the communication with this module is done through UART or RS232 Interface. The data is sent to the module or received from the module through UART interface.

The module is typically connected to +4.0V standard power supply. It can work on +4.5V regulated power and any higher voltage may damage the module. And the power source should be able to deliver a peak current of 2A. The UART interface is established as shown in figure.All you need to do is connect RXD of module to TXD of Arduino and TXD is connected to RXD of ARDUINO. The ground of controller and module must be connected for voltagereference. Here AUDIO IN is connected to MIC and AUDIO OUT is connected to a speaker or headset. And at last we need to connect a working GSM SIM card to the module. On powering the module, the NETLIGHT LED will blink periodically to state successful connection.

After all connections are done, we need to write a program for the microcontroller to exchange data with module. Since data exchange sequence between controller and module is really complex we will use libraries prewritten for the module. You can download libraries for controller or module through their websites. Using these libraries makes the communicationeasy. All you need to do is download these libraries and call them in programs. Once the header file is included, you can use simple commands in the program to tell the controller to send or receive data. The controller sends the data to the module through UART Interface based on protocol setup in libraries. The module sends this data to another GSM user using cellular network. If the module receives any data from the cellular network (or another GSM user) it will transmit it to controller through UART serial communication.

This way we can use GSM900A module to establish cellular connection.

Applications:

Cellular Communication

Robotics

Mobile Phone Accessories

Servers

Computer Peripherals

Automobile

USB Dongles

4.5 MOTOR DRIVER:

TB6600 Stepper motor driver shown in fig 4.5.a is a cheap and effective device which provides micro stepping ability to a stepper motor. It is suitable for driving 2 phase and 4 phase hybrid stepper motors. The motor driver compatible with any microcontroller providing a 5V signal

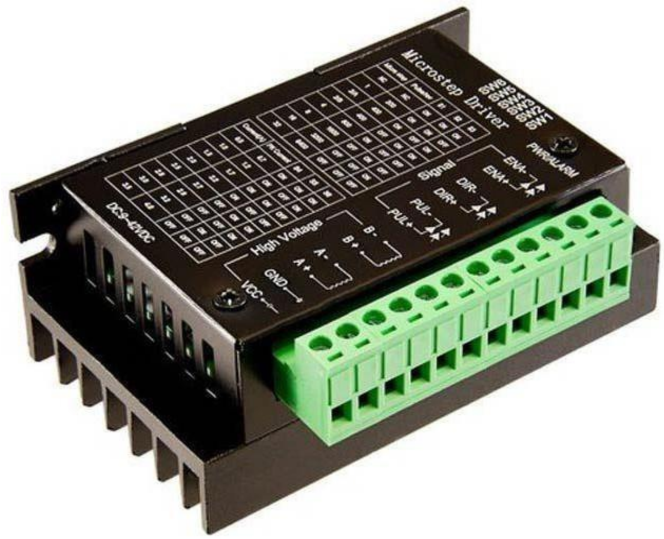


Fig 4.5.a TB6600 Stepper Motor Driver Module

Features and Specifications of TB6600 Stepper Motor Driver Module

This section mentions some of the applications of TB6600 stepper motor driver.

Operating voltage: 9-40V DC

Output Current: 0.7-4.0A

Pulse input frequency up to 20kHz

5V levels input signal

200-6400 pulse per revolution

Logic signal current: 8-15mA

Output current selectable in 8-steps via DIP switches

Suitable for 2 and 4 phase motors

Over current and over heat protection

Inputs are optically isolated

Fig 4.5.b TB6600 Stepper Motor Driver Module Pinout

TB6600 Pin Configuration/Pin Diagram

The TB6600 stepper motor driver module pin configuration/pin diagram Fig 4.5.b is describedbelow. There are 2 types of TB6600 driver modules such as 4A type and 4.5A type as shown below, with similar functions and pin configuration.

Pin 1: ENA- (ENA): This pin refers to the negative enable pin of the module.

Pin 2: ENA+(+5V): This pin refers to the positive enable pin, which is the +5V pin of themodule.

Pin 3: DIR-(DIR): This pin refers to the negative direction of the motor.

Pin 4: DIR+(+5V): This pin refers to the positive direction of +5V.

Pin 5: PUL-(PUL): This pin refers to the negative pulse. (For controlling rotations stepsof the motor)

Pin 6: PUL+(+5V): This pin refers to the positive pulse of the motor.

Pin 7: B-: This pin refers to the negative lead stepper motor coil wire 2.

Pin 8: B+: This pin refers to the position lead of stepper motor coil wire 2.

Pin 9: A-: This pin refers to the negative lead of stepper motor coil wire 1.

Pin 10: A+: This pin refers to the positive lead stepper motor coil wire 1.

Pin 11: GND: This pin refers to the common ground connection of the module.

Pin 12: VCC: This pin refers to the input supply voltage for the stepper motor drivermodule, which is 9V-42V.

The in-built control/driver switches SW1, SW2, SW3, SW4, SW5, and SW6 are used to control the resolution of micro steps and used to limit the driver current. Switching of SW1, SW2, and SW3 changes the micro-step resolution from full step to 1/32 step.

Switching of dip switches SW4, SW5, and SW6 ON Or OFF adjusts current (0.7 to 4Amps) during the continuous motor run.

TB6600 Stepper Motor Driver Technical Specifications:

The technical specifications of the TB6600 stepper motor driver module are given below.

The operating voltage ranges from 9-40V DC.

The output current of the module is 0.7A-4.0A and it is selected in 8 steps through DIPswitches
The input pulse frequency is up to 20KHz.

Input signal suitable for – 5V signal levels.

Pulse per revolution is 200-6400.

The logic signal current is 8A-15A.

It is suitable for 2-phase and 4-phase stepper motors.

It provides protection from overcurrent over-heat.

Inputs are isolated optically.

Insulation resistance is 500 megohms.

It can support PUL/FIR mode.

It is low cost.

4.6 STEPPER MOTOR

Fig 4.6a NEMA 17 Stepper Motor

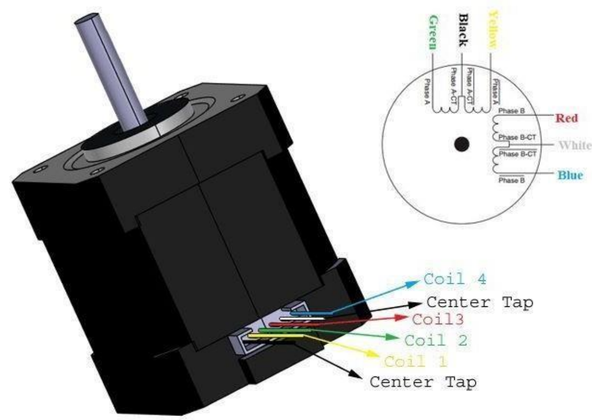


Fig 4.6.b NEMA 17 Stepper Motor Pinout

NEMA 17 Fig 4.6.1 is a hybrid stepping motor with a 1.8° step angle (200 steps/revolution).Each phase draws 1.2 A at 4 V, allowing for a holding torque of 3.2 kg-cm. NEMA 17 Stepper motor pinout Fig 4.5 is generally used in Printers, CNC machines and Laser Cutters.

NEMA 17 Stepper Motor Technical Specifications

Rated Voltage: 12V DC

Current: 1.2A at 4V

Step Angle: 1.8 deg.

No. of Phases: 4

Motor Length: 1.54 inches

4-wire, 8 inch lead

200 steps per revolution, 1.8 degrees

Operating Temperature: -10 to 40 °C

Unipolar Holding Torque: 22.2 oz-in

NEMA17 Stepper Motor is commonly used in CNC machines, Hard Drives and Linear Actuators. The motor have 6 lead wires and rated voltage is 12 volt. It can be operated at lower voltage but torque will drop. These motors has a step angle of 1.8 deg., this means that it has 200steps per revolution for every step it will cover a 1.8° hence the level of control is also high.

These motors run on 12V and hence can provide high torque. So if you are looking for a compacteasy to use stepper motor with high torque then this motor is the right choice for you.

Stepper Motor Applications

CNC machines

Precise control machines

3D printer/prototyping machines (e.g RepRap)

Laser cutters

Pick and place machines

When DC voltage is applied to the terminals of brushed DC motors, the motors continue to spin.Stepper motors are well known for their ability to transform input pulse trains (usually composedof square waves) into accurate increments in rotational position. Each pulse rotates the shaft through a fixed angle. A stepper motor Fig 4.6.3 consists of several "toothed" electromagnets arranged as a stator revolving around a central iron rotor. One electromagnet is powered to magnetically attract the gear teeth and turn the motor shaft. When lined up with the first electromagnet, the gear teeth areslightly offset from the next electromagnet. By turning off one electromagnet and turning on the next, the gear spins a little to line up with the next electromagnet. The process is then repeated.In a stepper motor, a full rotation is made up of a precise number of precise steps, eachcalled a 'step".

APPLICATIONS:

3D PRINTER:

A stepper motor is almost present in lists of common 3D printer parts. This is because a stepper motor is a highly accurate and economical
means to make extremely exact rotations and actions while a 3D printer is attempting to convert data from digital scans into actual 3D objects.

COMPUTER NUMERICAL CONTROL (CNC):

The majority of CNC machinery can be powered by stepper motors instead of servo motors. Numerous manufacturing processes use CNC applications, in which machine tools are operated and moved physically by pre-programed computer software in manufacturing and fabrication environments.

CAMERAS:

For use in camera and video surveillance positioning systems, stepper motors also offer severalother desirable characteristics, such as full
torque at a standstill, extremely quick and accurate response times for all movement inputs, consistent repeatability of predetermined
movements, and simple open-loop controls defined by fixed step sizes.

4.7 LCD

Fig 4.7 20*4 LCD Display

Description

This 20x4 Character LCD Display is built-in with RW1063 controller IC which are 6800, 4 lineSPI or I2C interface options. The WH2004G 20x4 LCD Display have the same AA size and pinassignment as existing WH2004A and WH2004B character LCD modules but with smaller outline and VA size. Below are the available series model numbers

WH2004G - 6800 interface

WH2004G1 – 4 line SPI interface

WH2004G2 - I2C interface

A 20x4 LCD (Liquid Crystal Display) is a type of alphanumeric display commonly used invarious electronic devices, such as digital clocks, calculators, and other similar devices. It consists of a total of 20 columns and 4 rows of characters, allowing it to display up to 80 characters at once.

The display uses a backlight to illuminate the characters on the screen, and each character is made up of 5x8 pixels. The pixels are controlled by a microcontroller, which sends signals to theLCD to turn on or off the pixels as needed to display the desired characters.

One advantage of using a 20x4 LCD display is that it allows for more text to be displayed thansmaller displays, making it ideal for applications where a lot of information needs to be shownon a single screen. Additionally, because it is a common display size, there are many libraries and resources available for programming and interfacing with the display.

However, a potential disadvantage of using a 20x4 LCD display is that it can be more expensiveand larger in size than smaller displays, which may not be suitable for certain applications wherespace and cost are a concern.

Overall, a 20x4 LCD display is a versatile and widely-used display option that can be useful for avariety of applications.

Here are some general specifications for a 20x4 LCD display:

Display type: Alphanumeric LCD display

Screen size: 20 characters x 4 lines

Display resolution: 5 x 8 dots per character

Character size: 2.95mm x 4.75mm

Viewing area: 76mm x 25.2mm

Backlight: White LED (or other color)

Operating voltage: 5V DC

Operating temperature range: -20°C to 70°C

Interface: 4 or 8-bit parallel interface

Controller: HD44780 (or compatible)

Contrast control: Via onboard potentiometer

Power consumption: Typically around 1mA for the display only, plus additional currentfor the backlight (varies by manufacturer)
Character set: ASCII (or other character sets)

It's important to note that specific specifications may vary depending on the manufacturer and model of the LCD display.

4.8 BVM BAG (AMBU)

High quality design used worldwide, established for many years, the Inter surgical BVM resuscitator is used worldwide, as an essential piece of emergency resuscitation equipment. Designed for ease of use, the Inter surgical BVM resuscitator is available in a full range of sizes: Adult (1.5 liter), Adult (1 liter), Pediatric (550ml) and Infant (280ml) sizes. The high quality, robust construction provides the reliability needed to withstand use in all scenarios. Supplied as afully assembled single use product, the BVM resuscitator is always ready for use in an emergency situation, thus avoiding the issues associated with reusable BVMs, including the risk of cross infection and the potential for components to be lost or misassembled following reprocessing.

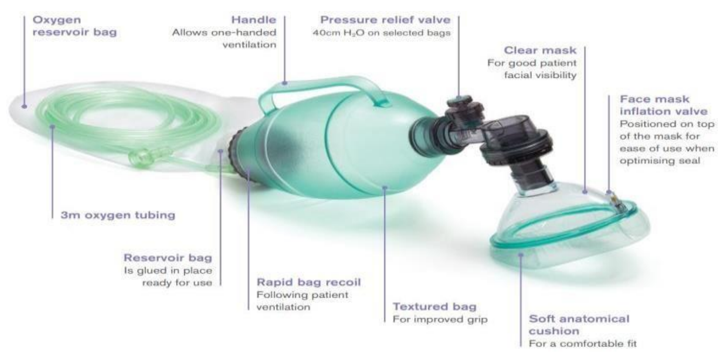


Fig 4.8 Bag Valve Mask

1. The textured bag with integrated handle improves grip for easier use in scenarios where only one user is present. Excellent recoil properties ensure the bag is ready for subsequent patient ventilations, and its transparency allows a clear view of the product and valves in use.
2. All Inter surgical BVM resuscitators have three meters of kink resistant oxygen tubing as standard and

a 360º swivel between the mask and valve to allow the BVM resuscitator to be heldin a convenient position.

3. The product packaging is fully flexible which doesn't restrict storage in areas where space is limited and 40cm H2O pressure relief valves are featured on the 1 litre, 550ml and 280ml BVM resuscitators.

4. The clear face mask allows excellent patient visibility and features a valve to optimise the seal,which is positioned on the top of the mask for ease of use. A standard 15F/22M connector allowsthe attachment of alternative face masks and endotracheal tubes or supraglottic airways asrequired.

5. A PEEP valve adaptor is available which allows the attachment of a PEEP valve to the BVM resuscitator.

4.9 FLOW SENSOR

Fig 4.9 Flow Sensor (YF-S201)

Accurate flow measurement is an essential step both in the terms of qualitative and economic points of view. Flow meters have proven excellent devices for measuring oxygen flow, and now it is very easy to build a water management system using the renowned oxygen flow sensor YF- S201. For example, you can make a robotic cocktail dispensing machine, and can use this sensor to accurately measure components like soda, water, air etc.

YF-S201 1/2-inch Flow Sensor sits in line with the water line and contains a pinwheel sensor to measure how much water has moved through it. There is an integrated magnetic Hall-Effect sensor that outputs an electrical pulse with every revolution. YF-S201 1/2-inch oxygen flow sensor has only three wires and it can be easily interfaced between any microcontroller and Arduino board. It requires only +5V Vcc and gives pulse output, the sensor needs to be tightly fitted between water pipeline.

4.10 ARDUINO IDE

A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a developmentenvironment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio(older) and Atmel Studio (newer).

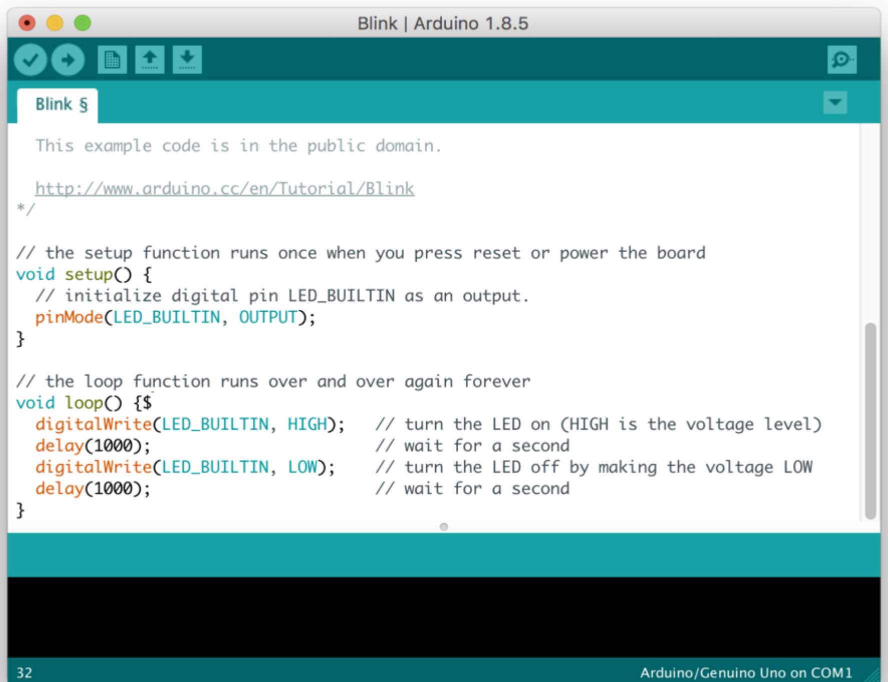


Fig 4.10 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application(for Microsoft Windows, macOS, and Linux) that is written in the Java programming language. It originated from the IDE for the languages Processing and Wiring. It includes a code editorwith features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the 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 thesketch 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. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

From version 1.8.12, Arduino IDE windows compiler supports only Windows 7 or newer OS.On Windows Vista or older one gets "Unrecognized Win32 application" error when trying to verify/upload program. To run IDE on older machines, users can either use version 1.8.11, or copy "Arduino-builder" executable from version 11 to their current install folder as it's independent from IDE.

CHAPTER 5
INTERFACING AND TESTING

5.1 MAX30100 WITH ARDUINO NANO

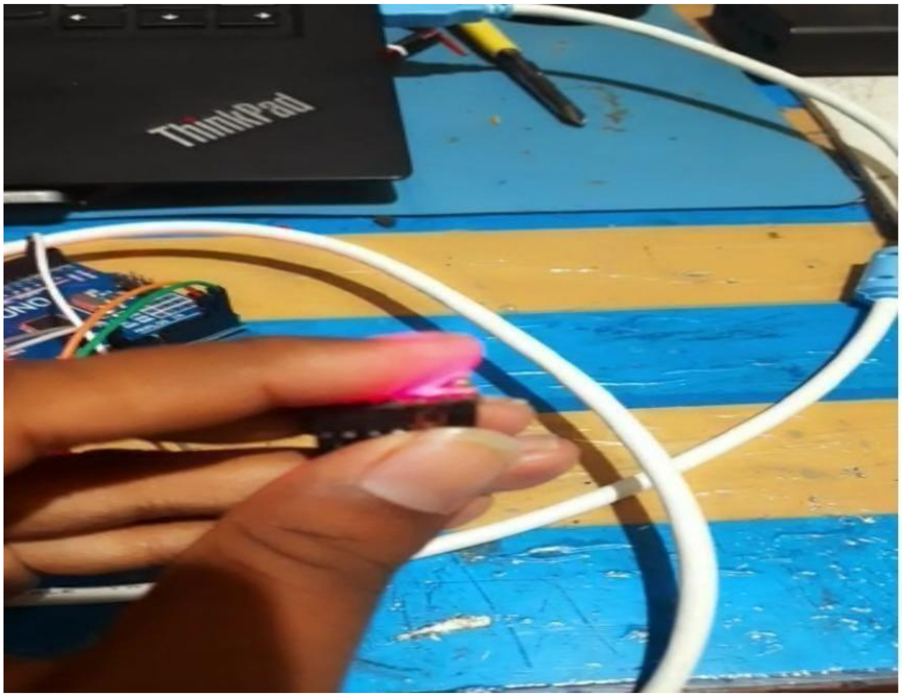


Fig 5.1.a Spo2 working

In this project we used max30100 and it is also known as spo2 sensor which is used to check the oxygen value in our body. once it check's a oxygen value in our body it will display the valuein the lcd using microcontroller.

The MAX30100 is a pulse oximeter and heart-rate sensor module that uses infrared and red light to measure the oxygen saturation level and heart rate of a person's blood. It can be used in conjunction with an LCD display to display the measured data in real-time.

- To use the MAX30100 with an LCD display, you will need a microcontroller or development board, such as an Arduino, to interface between the two devices. Here are the basic steps for connecting and using the MAX30100 with an LCD display:

- Connect the MAX30100 module to the microcontroller or development board using the I2C interface. The module will typically have pins for SDA (data) and SCL (clock), as well as VCC and GND.

- Connect the LCD display to the microcontroller or development board using a compatibleinterface, such as a parallel or serial interface. The display will typically have pins for power (VCC and GND), data (RS, RW, EN, and D0-D7), and backlight (if applicable).

- Write a program on the microcontroller or development board to read the data from the MAX30100 and display it on the LCD display. This will typically involve initializing the MAX30100 module and configuring it to read the desired data (such as heart rate or oxygen saturation level), as well as sending the data to the LCD display to be displayed.

- Test the system to ensure that the data is being read correctly from the MAX30100 and displayed on the LCD display in real-time.

Overall, using the MAX30100 with an LCD display can be a useful way to display the measured data in real-time and provide visual feedback to the user.

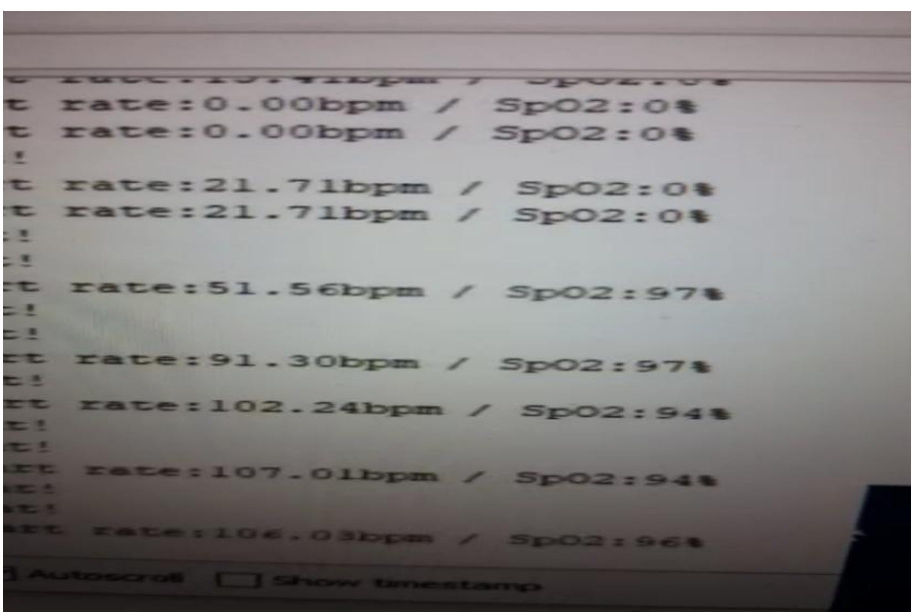


Fig 5.1.b Oxygen Value

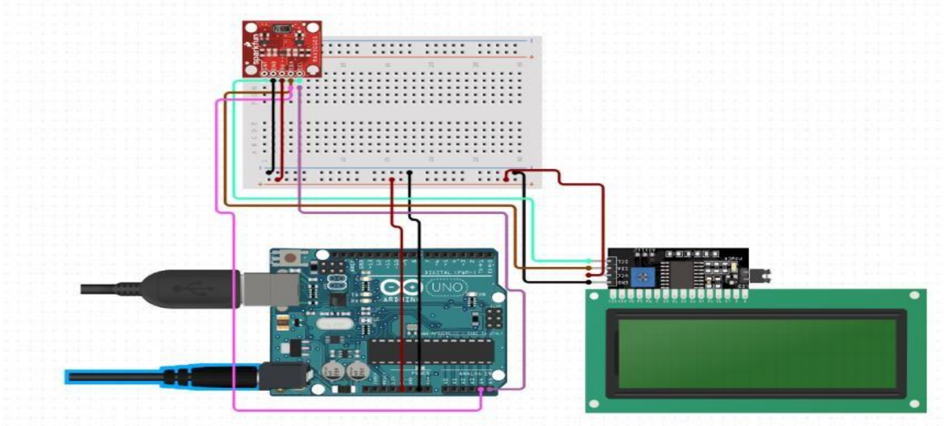


Fig 5.1.c Circuit Diagram

5.2 STEPPER WITH DRIVER USING ARDUINO

In our project we used nema 17 stepper motor to generate the oxygen from the bag valve mask. When the stepper motor rotates the pinion rotates together therefore the jack also moves forward near the bvm which compress the self-inflating bag of a bvm by which the oxygen obtained.

Here are the basic steps to control a NEMA 17 stepper motor using an Arduino and a stepper motor driver module:

- Connect the stepper motor to the driver module. The NEMA 17 stepper motor typically has four wires, which are connected to the driver module's four output pins. The wire colors may vary depending on the motor and driver module used, so refer to the datasheets for the specific components to determine which wire goes to which pin.

- Connect the driver module to the Arduino. The driver module typically has four input pins that are connected to the Arduino's digital output pins. The input pins are typically labeled STEP, DIR, EN (or ENABLE), and GND. Connect the STEP pin to a digital output pin on the Arduino (e.g. Pin 2), the DIR pin to another digital output pin (e.g. Pin3), and the EN pin to another digital output pin (e.g. Pin 4). Connect the GND pin to aground pin on the Arduino.

- Load the Stepper library in the Arduino IDE. This library provides functions for controlling the stepper motor using the driver module.

• Write a program to control the stepper motor. This program will typically involve initializing the stepper motor object with the correct number of steps per revolution, setting the motor speed and direction, and calling the stepper motor object's step() function to move the motor

• Upload the program to the Arduino and test the motor. The motor should rotate one full revolution in one direction and then stop for 1 second before rotating again. If the motoris rotating in the wrong direction, swap the connections to the DIR pin on the driver module.

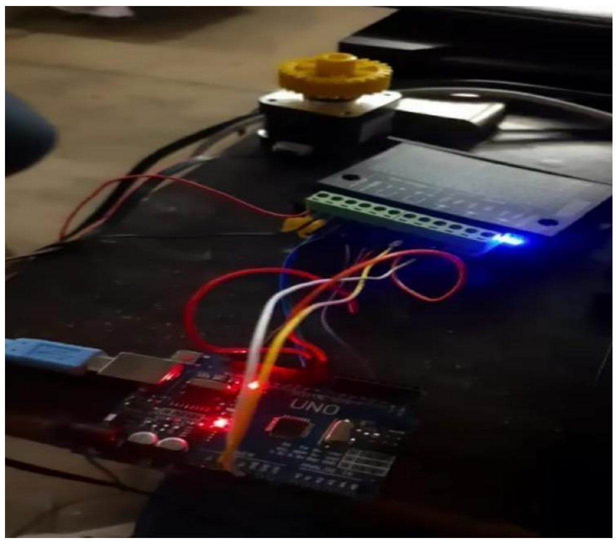


Fig 5.2.a Stepper motor working

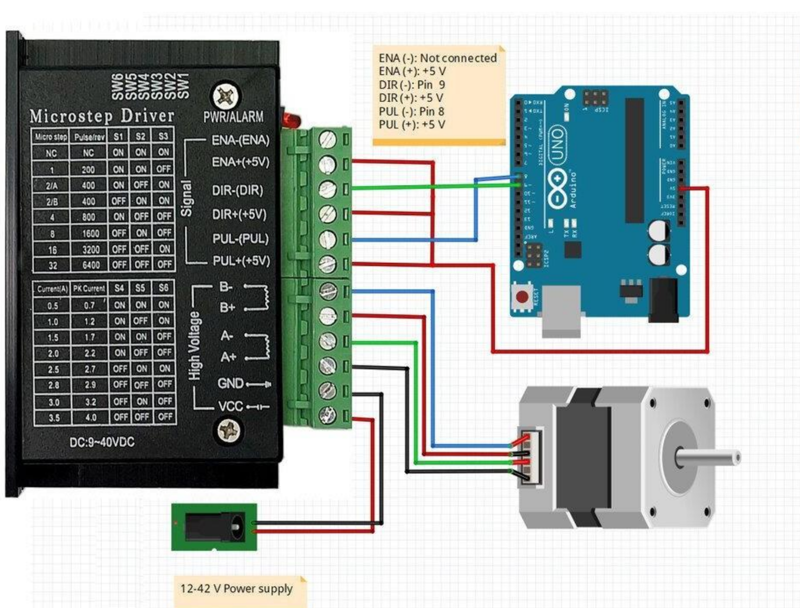


Fig 5.2.b Circuit Diagram

5.3 GSM WORKING

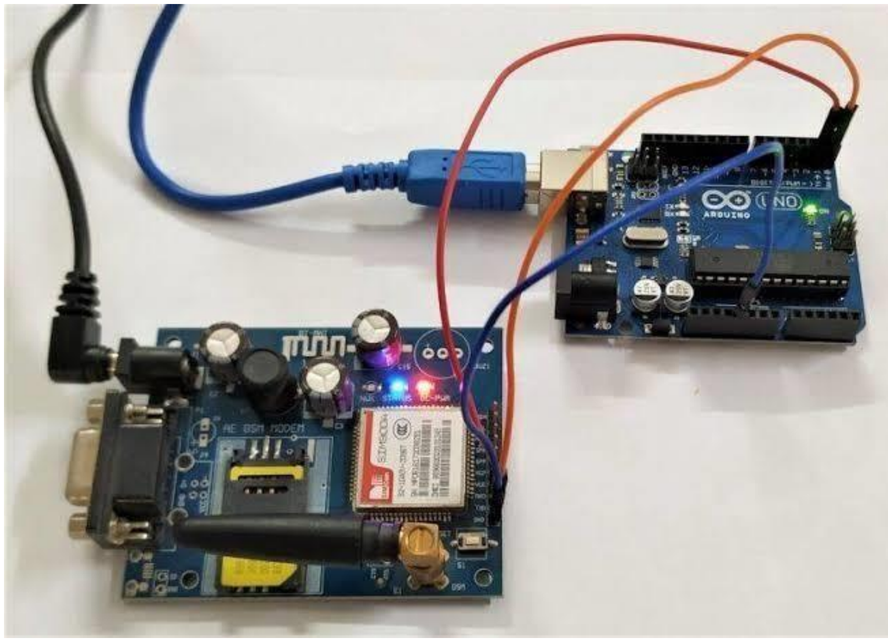


Fig 5.3.a GSM working

In this project we use GSM for wireless communication purpose , it sends a information of a patient (which contains the value of o2) to an attender/doctor who is caretaker of a particular patient when it is emergency via message

The SIM900 GSM module can work with Arduino to enable communication over the mobile network. Here are the basic steps to use the SIM900 GSM module with Arduino:

Connect the SIM900 module to the Arduino: Connect the TX pin of the SIM900 module to the RX pin of the Arduino, and the RX pin of the SIM900 module to the TX pin of the Arduino. Connect the GND pin of the SIM900 module to the GND pin of the Arduino, and the VCC pin of the SIM900 module to a 5V power source.
Install the GSM library: In the Arduino IDE, go to "Sketch" > "Include Library" > "Manage Libraries" and search for "GSM". Install the "GSM" library provided by Arduino.

Initialize the GSM module: In your Arduino sketch, include the GSM library, initializethe GSM module by calling the GSM.begin() function, and set the PIN code for the SIM card by calling the

GSM.begin(pinCode) function.

Send and receive SMS messages: Once the module is initialized, you can use the GSM_SMS library to send and receive SMS messages. To send an SMS message, use thesms.beginSMS() function to specify the recipient's phone number, the sms.print() function to write the message, and the sms.endSMS() function to send the message. To receive SMS messages, use the sms.available() function to check for incoming messages and the sms.read() function to read the message.

Make calls: You can use the GSMVoiceCall library to make phone calls with the SIM900module. To make a call, use the voiceCall.dial(number) function to specify the phone number to call.

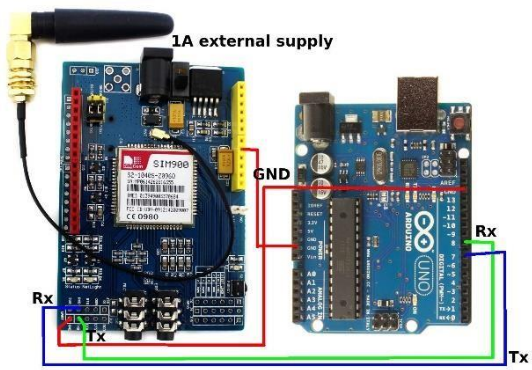


Fig 5.3.b Circuit diagram

5.4 FLOW SENSOR WITH ARDUINO

Fig 5.4.a Flow sensor with BVM

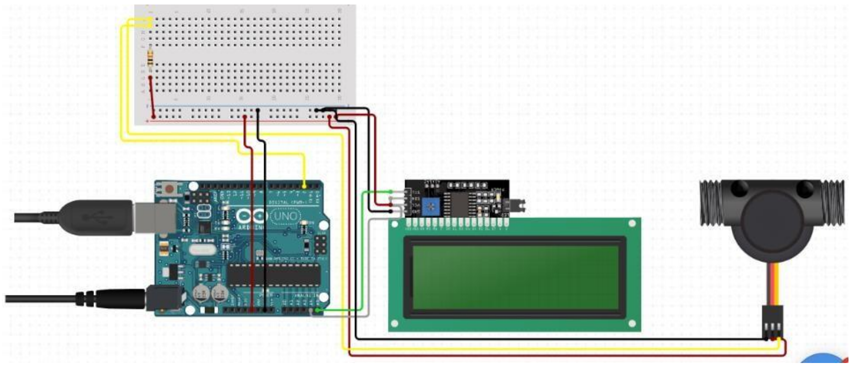
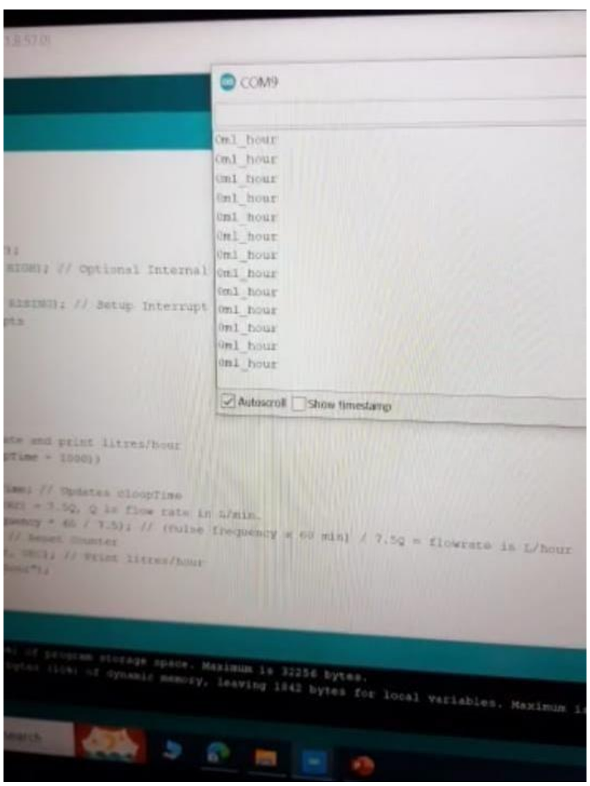


Fig 5.4.b Circuit diagram

In this project we used YF-S201which is an flow sensor used as an air flow sensor which detectsthe flow of oxygen which is obtained from the bag valve mask and shows the amount of oxygen flow's through it in the lcd display.



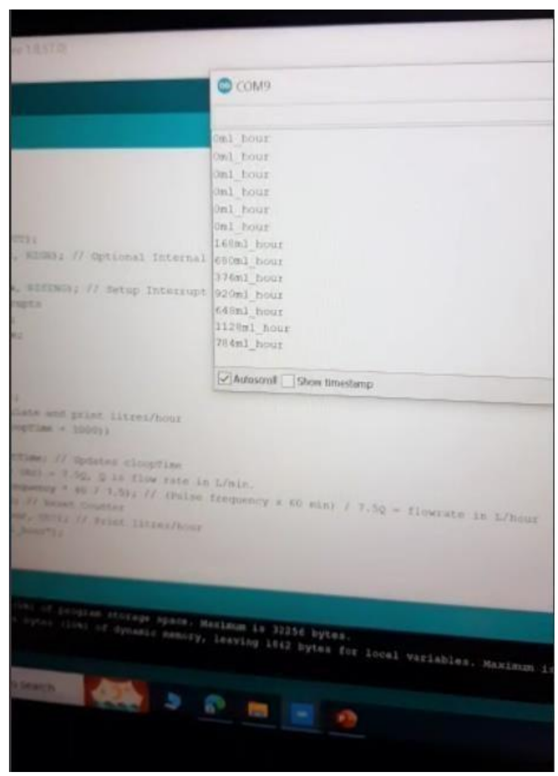


Fig 5.4.c Initial values

Fig 5.4.d Final Values

The YF-S201 flow sensor can be used with an Arduino board to measure liquid flow rate. Here are the basic steps to use the YF-S201 flow sensor with Arduino:

Connect the YF-S201 flow sensor to the Arduino: The YF-S201 flow sensor usually has three pins: VCC, GND, and OUT. Connect the VCC pin of the sensor to the 5V pin of theArduino, the GND pin of the sensor to the GND pin of the Arduino, and the OUT pin of the sensor to any digital pin of the Arduino, such as pin 2.

Install the PulseIn function: In the Arduino IDE, go to "Sketch" > "Include Library" > "Manage Libraries" and search for "PulseIn". Install the "PulseIn" library provided by Arduino.

Write the code: In your Arduino sketch, use the pulseIn() function from the PulseIn library to measure the duration of the pulse generated by the YF-S201 flow sensor. The pulse duration is proportional to the flow rate of the liquid. You can convert the pulse duration to flow rate by calibrating the sensor with a known flow rate.

We use the attach Interrupt() function to count the number of pulses received from the flow sensor. We detach the interrupt, calculate the flow rate, and then attach the interrupt again to continue counting pulses.

Note that the calibration factor of 7.5 may vary depending on the specifications of your YF-S201 flow sensor and the liquid being measured. You may need to adjust this value tomatch your specific use case.

5.5 BVM BAG WITH STEPPER

The below figures show's the working of BVM bag where pumping operation is done using stepper motor.

The pumping action of the Ambu bag system is done by the stepper motors attached with a scotch yoke like mechanism connected to a stepper motor. The function of the scotch yoke mechanism is to convert the rotatory motion of the stepper motor into the linear push pull motion. The stepper motor is connected to the motor driver, the motor driver delivers enough power to the motor to rotate. In this system two motor are used so that the dual ventilation is possible at a time.

The stepper motor is connected to the motor driver, the motor driver delivers enough power to the motor to rotate. In this system two motor are used so that the dual ventilation is possible at a time. The pressure of the pumping is controlled using a pressure varying knob(potentiometer). By varying knob, we can control the speed of the motion of the motor and thus by the speed of the pressure exerted by the scotch yoke mechanism. A pressure sensor BMP280 is also used here in the Ambu bag to measure the pressure of Ambu bag.

When a patient is not breathing, the Ambu bag is used in place of mouth-to-mouth resuscitation. This might happen in situations such as drowning, cardiac arrest, or a drug overdose. Its use requires training, typically to make sure the patient is getting enough oxygen when it is being used. Other problems which may result in further injury or even death can occur when using the resuscitator, such as damage to the throat, forcing of air into the stomach, etc. So, the use of mechanical portable ventilator is highly technical and complicated. It should not be attempted by anyone who is not fully trained. This training is only provided to first responders and medical professionals.

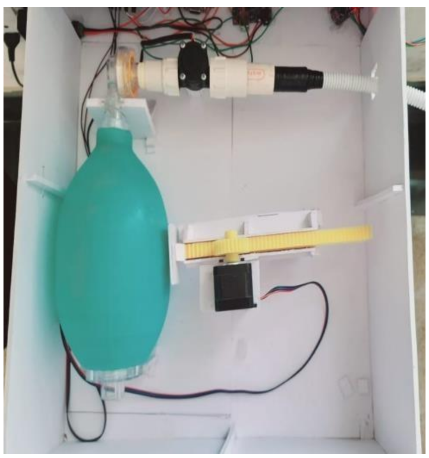


Fig 5.5.a Initial Stage

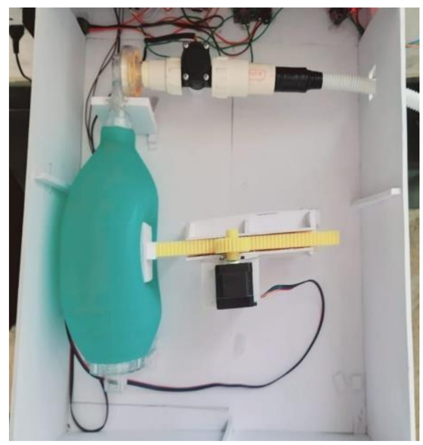


Fig 5.5.b Final Stag

\

CHAPTER 6

RESULTS & DISCUSSIONS

The below figures 6.1, 6.2 and 6.3 represents the prototype of this project.

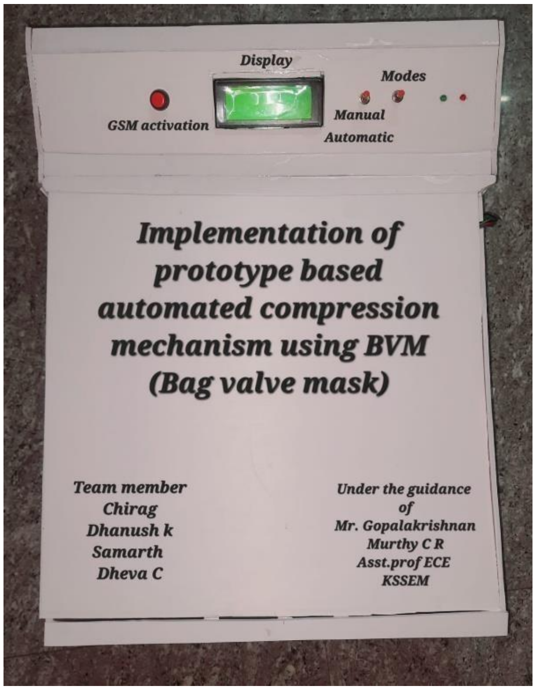


Fig 6.1 Prototype Front View

Fig 6.2 Inside View I

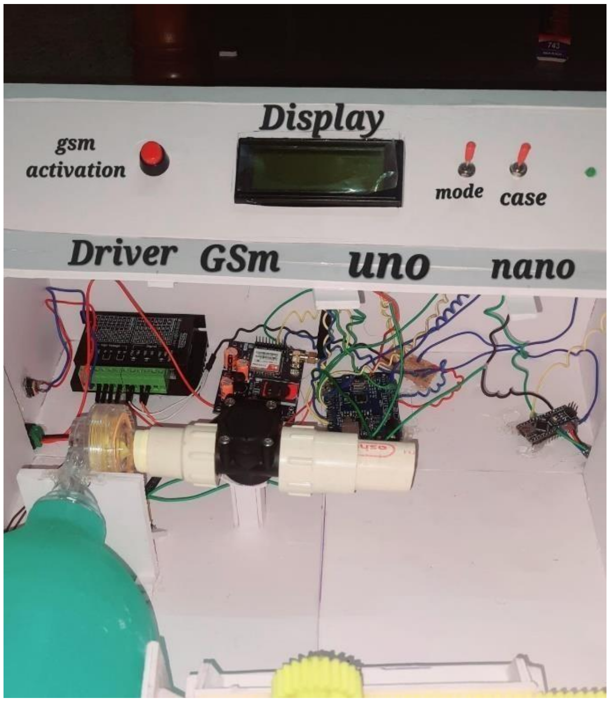


Fig 6.3 Inside View II

Fig 6.4 Spo2

The above figure represents the spo2 sensor in the prototype where connected to Arduino Nano displaying in lcd.

The prototype contains two section one patient side and the other one is care taker or doctor side,the data transmission is wireless. The patient side consists of sensor, oxygen sensor to measure the oxygen value of the patient (Spo2) and GSM to know the condition of the patient. Oxygen sensor is used to monitor heart rate regularly and accurately. The values from the oxygen sensor are periodically sent to GSM Once the data is received by the GSM it is successfully transmitted to the concerned caretaker or doctor which is monitored with less delay and also data showed on application. The heartbeat of the normal person varies between 60 to100 beats per minute.

The patient side will have a mechanical ventilator in case of emergency the patient can be supplied the oxygen and the care taker or doctor side will get a indication from the LED saying alert. The exact output of the system is we can track and monitor the patient health. We can trackthe patient with the help of GSM implementation. We can also monitor the patient health parameter using various sensors. The mechanical ventilator can turned on or off by the application when the oxygen level goes low or when there is need. The care taker or the doctor side will have an application in the mobile where data can be viewed and a GSM module whichis used for the data transmission. SMS application is an Iot platform used to send and view the data received at the care taker or doctor side which is successfully sent. This platform will represent the uploaded values in digital number which makes analysis easier.

This system helps us to decrease the death rate of the patient by helping the caretaker or doctor with important information and alert do that the patient will be taken care in time. The technology is helping one step closer in order to increase the care of the patient.

Fig 6.5 Automatic Mode (Normal Condition)

The figure above represents an Automatic Mode Normal Condition in which a patient's breathing is normal. In this mode, the ventilator is set to an Automatic Mode in which it automatically adjusts the pressure provided to the patient based on their level of breathing, with the goal of providing a consistent and comfortable level of support. This mode is commonly used when a patient is stable and not requiring much intervention in terms of their respiratory requirements. The pressure provided by the device will adaptively increase and decrease to ensure a sufficient level of oxygen is provided to the patient, while maintaining a comfortable and safe level of support. This mode can provide a way to ensure that a patient's respiratory needs are being met without intensive medical intervention. As such, it is sometimes used when a patient is stable, but must remain under close supervision. Additionally, this setting may be used in situations where a patient is unable to control their breathing, such as those with sleep apnea or other serious respiratory issues.

Fig 6.6 Automatic Mode (Abnormal Condition)

The figure 5.6 above represents an Automatic Mode Abnormal condition, where the patient is in a critical condition. The patient requires more oxygen than normal to be supplied through the ventilator in order to help stabilize their condition. Severe distress and fatigue can be the result of not having enough oxygen, and with the Automatic Mode Abnormal condition, it has become even more of a struggle for the patient. As well, it is important to note that the patient can recall events that would have damaged the brain. Outcomes of this condition can be very serious, and the patient requires constant monitoring and support. The long term outlook of the condition is often unpredictable, and can lead to a severe chronic condition. Therefore, the patient needs to be treated in a specialized manner. Ventilation, oxygen, fluid management, and other medications might be necessary for the patient to survive. In short, the Automatic Mode Abnormal condition is a serious and potentially life-threatening condition which needs to be monitored carefully and treated aggressively in order to maintain the patient's health and well-being.

Fig 6.7 Manual Mode (Cases)

The above figure represents the manual mode for providing oxygen to the patient according to their age. There are two cases in the manual mode, Case 1 and Case 2, which each represent different conditions of Inhale to Exhale ratio. Case 1 represents a condition of Inhale to Exhale ratio of 1:2 and Case 2 represents a condition of Inhale to Exhale ratio of 2:1. According to the age of the patient, oxygen is provided in milliliters. This manual mode of providing oxygen is beneficial for hospitals with low budgets that cannot afford the proper CO2 monitors in order for automating the process. This manual mode also provides a sense of control to the healthcare professionals, allowing them to have an even more fine-tuned approach to care for the patients according to their individual needs. Moreover, manual mode also allows the healthcare professionals to modify the oxygen flow according to the clinical state of the patient, thus increasing the comfort of the patient and reducing the chances of any medical complications.

CONCLUSION AND FUTURE SCOPE

CONCLUSION

In this proposed project, a prototype device to assist the patients who can partially breathe by their own is developed. The motivation come from world-wide shortage of mechanical ventilators in the treatment of covid-19 patients. This device is provided with very basic design and reliable structure that is easily acceptable by the patient. Main focus in this model is to minimize the components and increase its efficiency, so that while using this device to the patient, they should feel as comfortable as the normal ventilator. The biggest advantage of our system is that it includes, with two operating modes one is automatic mode and another one is manual mode. Our system is capable of providing ~500ml of oxygen to the patient at any time and we have set the breathing cycle if the oxygen level is normal condition the system will pump at ~16Rr per minute and when the oxygen level is below 85% it will automatically pump at ~22Rr per minute oxygen level in blood is monitored using the spo2 sensor and it is displayed in the monitor and through GSM it send the SMS to the intended person if the oxygen level is below 85% and the doctor has to switch on the gsm to activate the gsm after sending the message to ensure they have treated the person, the doctor has to switch from automatic to manual by switching the toggle switch, in manual mode we have two cases using the we can change the breathing time like inhale and exhale ratio now we have fixed two cases one is 1 second inhale and 2 second exhale and in another case 2 second inhaling and 1 second exhaling and we have achieved in our project.

FUTURE SCOPE

• Alert system: Any low level of oxygen can be detected using a Buzzer. A built-in alert system can be embedded to active and warn the local and remote locations about any malfunctioning for immediate attention.

• IoT based ventilator: The last few years have seen exponentially growth in wireless communication technologies with the emergency of 5G, artificial intelligence, robotic, cloud computing wireless sensor network. A ventilator can be monitor remotely by a mobile application using advance technology and IoT devices. And IoT based ventilator can be design to display output and parameters in applications using graph widgets which provide low latency, enhanced and fast service, location awareness notification services etc

• An alternator ventilator controlling system The IoT based ventilator can be embedded with PMW controlling system with the main control chip using ESP32. The use of ESP32 chip aims to control IoT based ventilator with Bluetooth communication so that distance can be maintained. In addition to the

IoT based monitoring and controlling functions. The ventilator can also be adjusted with several control buttons provided near the system.

• Safety mode: An alternator power source can be embedded so that when power goes down the back up battery automatically kicks in

REFERENCES

[1] Tassadaq Hussain, Amna Haider, "Synchronized Intermittent Mandatory Ventilation Mode Control Using Pulse Oximeter", International Conference on Computing, Mathematics and Engineering Technologies, 2018.

[2] Subha Hency Jose P, P.Rajalakshmy, "A Novel Methodology for the Design of a Portable Ventilator", International Journal of Innovative Technology and Exploring Engineering, 2019.

[3] Endi Sailul Haq, Devit Suwardiyanto, "Mechanical Ventilator Control System Using Low-Cost Pressure Sensors", International Conference on Computer Science, Information Technology, and Electrical Engineering, 2021.

[4] Rameshwar Bhaginath Kale, Shashank Kumar Singh, "DIY Ventilator using Arduinowith Blood Oxygen Sensing for Covid Pandemic", International Research Journal of Engineering and Technology, 2021.

[5] Abhishek Pandey, Aradhya Joshi, "An Introduction To Low-Cost Portable Ventiator Design", International Conference on Advance Computing and Innovative Technologies in Engineering, 2021.

[6] Joshua M. Pearce, "A review of open source ventilators for COVID-19 and future pandemics", F1000Research, 2020.

[7] Vishal Kumar, Ravinder Kumar, "Recent advances in low-cost, portable automated resuscitator systems to fight COVID-19", Health and Technology, 2021.

[8] Mashoedah1, Umi Rochayati, Indra Hidayatulloh, "IoT Enabled Ventilator Monitoring System for Covid-19 Patients", Journal of Physics: Conference Series, 2021.

[9] Jozef Ziv c ak, Michal Kelemen, "A Portable BVM-based Emergency Mechanical Ventilator", IEEE 19th World Symposium on Applied Machine Intelligence and Informatics, 2021.

[10] Ahmed Ibarra Abboudi, Abdelrahman Ibrahim Alhammadi, "Design and Implementation of Portable Emergency Ventilator for COVID-19 Patients", 2 Advances in Science and Engineering Technology International Conferences, 2022.

[11] Cong Toai Truong1, Kim Hieu Huynh1, Van Tu Duong, Huy Hung Nguyen, "Model-free volume and pressure cycled control of automatic bag valve mask ventilator", AIMS Bioengineering, 2021.

[12] Leonardo Acho, Alessandro N. Vargas, "Low-Cost, Open-Source Mechanical Ventilator with Pulmonary Monitoring for COVID-19 Patients", Department of Mathematics, Universitat

Politècnica de Catalunya (UPC), 08222 Terrassa, Spain, 2020.

[13] Vijay Jadav1, Nayan Machhi, “Review for Arduino based Portable Ventilator for COVID-19”, International Journal of Creative Research Thoughts, 2022.

[14] LIMING HAO1, YAN SHI, “Dynamic Characteristics of a Mechanical Ventilation System With Spontaneous Breathing”, 1School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China, 2019.

[15] Shao-Yung Lu1, Hau Lin, Hsu-Tah Kuo, Chien-Liang Wu, “Design and Study of a Portable High-frequency Ventilator for Clinical Applications”, 1Electrical and Computer Engineering Department, National Chiao Tung University, Taiwan 2 MacKay Memorial Hospital, Taipei, Taiwan, 2019.

[16] Rohan Lal Kshetry, Arnab Gupta, Somnath Chattopadhyaya, “Design and Analysis of a LowCost Electronically Controlled Mobile Ventilator, Incorporating Mechanized AMBUBag, for Patients during COVID-19 Pandemic”, Hindawi Journal of Healthcare Engineering, 2022.

[17] Edwin Calilung, Jason Española, Elmer Dadios, Alvin Culaba, “Design and Developmentof an Automated Compression Mechanism for a Bag-Valve-Mask - Based Emergency Ventilator”, IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, 2020.

[18] Aviral Chharia, Shivu Chauhan1, Shankhaneel Basak, Bikramjit Sharma, “NeT-Vent: Low-Cost, Rapidly Scalable and IoT-enabled Smart Invasive Mechanical Ventilator with adaptive control to reduce incidences of Pulmonary Barotrauma in SARS-CoV-2 patients”, 1 2nd Global Conference for Advancement in Technology (GCAT) Bangalore, India, 2021.

[19] Muhammad Jawad Ghafoor, Mustafa Naseem, Fahad Ilyas, “Prototyping of a Cost Effective and Portable Ventilator”, Department of Electrical Engineering, Information Technology University 6th Floor, Arfa Software Technology Park, Ferozepur Road,ahore, 2017.

[20] Abrar Shaikh, Akshaya Kumar, Amit Kumar, “Arduino Based Low-Cost portable ventilator for Covid-19 Pandemic”, International Journal of Creative Research Thoughts, 2022