Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY
(An Autonomous institution affiliated to VTU, Belgaum and Aided by Government of Karnataka, INDIA)
Near Jnana Bharathi Campus, Outer Ring Road, Mallathally, Bengaluru-560056.

PROJECT REPORT
on
"DESIGNING A PORTABLE VENTILATOR"

Submitted in partial fulfillment of the requirements for the award of the Degree

BACHELOR OF ENGINEERING
in
MEDICAL ELECTRONICS
by
VINAY KUMAR K DESHPANDE 1DA19ML400

Under the Guidance
of
Dr. D K RAVISH
Associate Professor
Department of Medical Electronics
Dr. AIT, Bengaluru - 560056
DEPARTMENT OF MEDICAL ELECTRONICS
Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY
MALLATHAHALLI, OUTER RING ROAD, BENGALURU – 560056

2021-22

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY
Mallathahalli, Bengaluru - 560056

Department of MEDICAL ELECTRONICS

Certificate

Certified that the project work entitled **"DESIGNING A PORTABLE VENTILATOR"**, carried out by **VINAY KUMAR K DESHPANDE (1DA19ML400),** Bonafede students of Dr. Ambedkar Institute of Technology, Bengaluru – 560056 in partial fulfillment for the award of Bachelor of Engineering in MEDICAL ELECTRONICS of the Visvesvaraya Technological University, Belagavi during the year 2021–2022. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements.

Signature of the Guide	Signature of the HOD	Signature of the Principal
(Dr. D K RAVISH)	**(Dr. SHANTHI K J)**	**(Dr. MEENAKSHI.M)**

Name of the Examiners **Signature with Date**

1.__________________

2. __________________

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

Mallathahalli, Bengaluru - 560056

Department of MEDICAL ELECTRONICS

Declaration

We,**VINAY KUMAR K DESHPANDE (1DA19ML400),** hereby declare that, the project work entitled **"DESIGNING A PORTABLE VENTILATOR"** is independently carried out by us at Department of MEDICAL ELECTRONICS, Dr. Ambedkar Institute of Technology, Bengaluru-560056, under the guidance of **Dr. D K RAVISH,** Designation, Department of MEDICAL ELECTRONICS, Dr. Ambedkar Institute of Technology. The Project work is carried out in partial fulfillment of the requirement for the award of degree of Bachelor of Engineering in MEDICAL ELECTRONICS during the academic year 2020-2021.

Place: Bengaluru — Name & Signature of students

Date: — VINAY KUMAR K DESHPANDE

ACKNOWLEDGEMENT

The sense of jubilation that accompanies the successful completion of this major project would be incomplete without mentioning and thanking all the people who played a vital role in the completion of this project by providing endless encouragement and support.

We would like to thank **Dr. M. Meenakshi,** Principal, Dr.AIT, who has always been a great source of inspiration while carrying out this project work.

We would like to convey our utmost gratitude to **Dr. Shanthi. K J**, H.O.D, Medical Electronics Department for providing a congenial environment and surrounding to work in.

We are highly indebted to our guide **Dr. D K RAVISH**, Assistant Professor, Department of Medical Electronics, Dr.AIT for constant guidance and support, as well as for providing necessary information regarding the Major project.

We would also like to thank all the teaching and non-teaching staff members of Department of Medical Electronics for their support during the course of this Major Project implementation.

Lastly, we would like to thank our parents and friends whose constant encouragement and support was crucial in execution and completion of this Major project.

NAME: VINAY KUMAR K DESHPANDE
USN:1DA19ML400

ABSTRACT

This study describes the event of a straightforward and easy-to-build moveable machine-driven bag valve mask (BVM) compression system, which, throughout acute shortages and provide chain disruptions will function a brief emergency ventilator. The revitalisation system relies on the Arduino controller with a period of time package put in on a mostly RepRap 3D printable constant quantity component-based structure. the value of the materials for the system is beneath $170, that makes it cheap for replication by manufacturers round the world. The device provides a controlled respiration mode with recurrent event volumes from a hundred to 800 millilitre, respiration rates from five to forty breaths/minute, and inspiratory-to-expiratory quantitative relation from 1:1 to 1:4. The system is meant for responsibility and quantifiability of measuring circuits through the utilization of the serial peripheral interface and has the flexibility to attach extra hardware because of the object-oriented algorithmic approach. Experimental results when testing on a man-made respiratory organ for peak breath pressure (PIP), vital sign (RR), positive end-expiratory pressure (PEEP), recurrent event volume, proximal pressure, and respiratory organ pressure demonstrate repeatability and accuracy exceptional human capabilities in BVM-based manual ventilation. Future work is important to any develop and check the system to form it acceptable for preparation outside of emergencies like with COVID-19 pandemic in clinical environments, however, the character of the planning is specified desired options are comparatively straightforward to feature victimization protocols and constant quantity design files provided.

CONTENTS

CHAPTER 1

INTRODUCTION

Human lungs use the reverse pressure generated by contraction motion of the diaphragm to suck in air for respiratory. A contradictory motion is employed by a ventilator to inflate the lungs by pumping kind motion.

A ventilator mechanism should be able to deliver within the vary of ten – thirty breaths per minute, with the power to regulate rising increments in sets of two. beside this the ventilator should have the power to regulate the air volume pushed into lungs in every breath. The last however currently the smallest amount is that the setting to regulate the time length for inhalation to exhalation quantitative relation. Apart from this the ventilator should be able to monitor the patient's blood chemical element level and exhaled respiratory organ pressure to avoid over/under gas pressure at the same time. The ventilator we tend to hear style and develop exploitation Arduino encompasses of these necessities to develop a reliable nevertheless cheap DIY ventilator to assist in times of pandemic.

We here use a silicon ventilator bag coupled driven by DC motors with two aspect push mechanism to push the ventilator bag. we tend to use switch for change and a variable pot to regulate the breath length and also the beats per minute BPM value for the patient. Our system makes use of blood chemical element sensing element beside sensitive pressure sensing element to observe the required vital organ of the patient and show on a mini screen. Additionally, an emergency buzzer alert is fitted within the system to sound an alert as shortly as any anomaly is detected.

The entire system is driven by Arduino controller to realize desired results and to help patients in COVID pandemic and alternative emergency things. Current pricing of commercial mechanical ventilators in low-/middle-income countries (LMICs) markedly restricts their availability, and consequently a considerable number of patients with acute/chronic respiratory failure cannot be adequately treated. Our aim was to design and test an affordable and easy-to-build noninvasive bilevel pressure ventilator to allow a reduction in the serious shortage of ventilators in LMICs.

Noninvasive mechanical ventilation (NIV) could be a wide used and accepted treatment for chronic metabolic process diseases and, in some cases, it's conjointly another to invasive ventilation choices for patients with acute metabolic process failure caused by a range of aetiologias. though positive pressure ventilation in low- and middle-income countries (LMIC) is most often provided invasively, the advantages of NIV are being progressively recognized. Indeed, for obvious reasons of value and easy use, NIV seems to be not solely a good, however conjointly a very appropriate approach to supply metabolic process support in patients living in developing low-income economies. this is often particularly relevant since in these regions the burden of essential health problem is giant, and is anticipated to extend with growing urbanization, rising epidemics and increasing access to hospitals. what is more, the elevated value of attention staffing, infrastructure desires and burdensome access to provides have hampered the event of totally equipped medical care units in LMICs. As a consequence, the demand for cost-efficient medical instrumentality, like mechanical ventilators, is probably going to greatly increase in those countries. Moreover, mechanical ventilators are expensive, that markedly restricts their handiness and, consequently, the power to adequately treat a major range of patients with each acute and chronic metabolic process failure in LMICs. These problems are all the additional evident in lightweight of the continuing coronavirus pandemic, wherever even industrialized economies are encountering vital shortages within the range of obtainable ventilators to satisfy the stress obligatory by this malady, specified handiness of noninvasive metabolic process support is also valuable sure as shooting patients or as a short-lived bridge.

Philanthropic donation of medical devices could facilitate in providing mechanical ventilators to non-resourced regions in LMICs, however these initiatives are fraught with sizeable limitations. Indeed, donation of commercially obtainable instrumentality is pricey and is barely partly effective since it's been reported that up to five hundredth of given devices become unusable thanks to lack of adequate maintenance and inability to get spare elements. additionally, donations are hardly property as a result of they need long-run commitments, e.g., to supply device mating. during this context, different solutions that are supported in-house producing of pressure support devices may cut back the intense shortage of ventilators in LMICs. consequently, the aim of this study was to style and take a look at a unique low-priced bilevel pressure support ventilator, and supply open access to the elaborate technical data, thereby letting free and unrestricted replication and implementation. to determine adequate performance of the device, bench testing was dispensed supported simulated patients with obstructive/restrictive diseases below well-controlled conditions, a standard wide accepted approach to check therapeutic devices for metabolic process support. Then, and following the prevailing literature, the model ventilator was tested in healthy volunteers subjected to obstructive-restrictive loaded respiration to mimic patients with metabolic process diseases requiring NIV.

CHAPTER 2

LITERATURE SURVEY-1

A Novel Low-Cost Ventilator for Use in a Worldwide Pandemic: The Portsmouth Ventilator Cole, Jacob H. MD, LT, MC, USN; Hughey, Scott B. MD, LCDR, MC, USN; Rector, Christopher H. BS, HM3, USN[2]; Booth, Gregory J. MD, LT, MC, USN December 2020

In late 2019, a novel coronavirus was identified in Wuhan province, China. The virus was later identified as severe acute respiratory syndrome coronavirus-2, or COVID-19, and is the cause of a current worldwide pandemic. Based on the initial Chinese data, 14–17% of hospitalized patients required supplemental respiratory support. Italy was one of the first western countries with widespread disease. Their critical care facilities appeared to carry an enormous burden of the patients, with an estimated 16% of actively infected patients requiring admission to an ICU for hypoxic respiratory failure from COVID-19.

This is important for the current crisis, because in the absence of definitive treatment, supportive mechanical ventilation for several days to weeks is the mainstay treatment for severe disease. Currently, there are approximately 62,000 full function ventilators in the United States, with 98,000 basic ventilators and 8,900 in the strategic reserve. The Centers for Disease Control and Prevention estimate that between 2.4 and 21 million Americans will require hospitalization. Based on the Italian data, the number of patients requiring ventilators will range between 1.4 and 31 patients per ventilator. The U.S. Department of Health and Human Services has already started to encourage rationing of ventilator use by eliminating elective surgeries. Recent studies similarly project ventilator shortage in the United States.

Because of this need, we sought to build a low-cost ventilator for use when surge demand exceeds current capacity. The requirements for this ventilator were as follows:

- 1) Components must be easily sourced “off-the-shelf” items that are available to the general public.
- 2) They must have “open-source” compatibility, so the design will be widely available and technically easy to build.

- 3) Must be able to tolerate a range of ventilation strategies to tolerate high airway pressures associated with acute respiratory distress syndrome (ARDS).
- 4) A cost containment strategy must be maintained to ensure the ventilator would not be cost-prohibitive.

Although many modern ICU ventilators use a turbine to drive pressure, other types have included a servo control valve, bellows, and pneumatic pressure chambers ([8,9]). Considering the technical complexity of the turbine and servo control ventilators, we believed either a below- or pneumatic-type ventilator would be the easiest to use with the requirements we established. Initial draft designs resulted in high confidence in the pneumatic model, which we pursued. We hypothesized that the Portsmouth Ventilator would be noninferior to the standard-of-care ventilators, while still meeting our requirements.

Materials and Methods

This ventilator follows the above design and uses a standard ventilator breathing circuit. It incorporates three solenoid valves (Charging, Inspiratory, and Expiratory represented by "C," "I," and "E," respectively, in the above diagram) controlled by a simple microcontroller-driven electronics circuit. The ventilator connects to pipeline gas supply to both air and oxygen. A simple gas blender merges air and oxygen, and can realistically deliver either 100% oxygen, room air (21% oxygen), or a 60% oxygen gas blend to the patient. The gas mixture is delivered to the charging chamber at 50–55 PSI by opening the charging valve. Using Boyle law relationship ($P_1V_1 = P_2V_2$), the chamber volume at high pressure is discharged into the breathing circuit and patient lungs at a lower pressure and a higher volume (**Fig. 1**).

Following inspiration, the inspiratory valve is closed and the expiratory valve is opened. The expiratory valve opens through a positive end-expiratory pressure (PEEP) valve, allowing PEEP (0–20 cm H_2O) to be delivered to the patient. The tidal volume on this device is adjusted by adding or removing expansion chambers. Each chamber generates approximately 45-mL additional tidal volume (**Table 1**). The respiratory rate and inspiratory time are set using attached controls. Additionally, if the patient is spontaneously breathing, the ventilator can detect a respiratory effort by measuring decreased airway pressure and augment by delivering a breath. Sensitivity to the patient's inspiratory effort is similarly adjusted with controls attached to the microcontroller.

Parameter	Range
Respiratory rate	4–30 RPM
Inspiratory time	0.5–7.5 s
Positive end-expiratory pressure	0–20 cm H_2O
Max plateau pressure	35 cm H_2O
Tidal volume	350–800 mL
Fio_2	21–100%

Table 1: Ranges and Values for Performance of the Ventilator

The breathing circuit is equipped with a 0.5-PSI (35 cm H_2O) pressure relief valve to limit airway pressures exceeding 35 cm H_2O and a negative pressure relief valve to allow for spontaneous room air inspiration at any point during the respiratory cycle, preventing a negative pressure injury (if the assist-control [AC] portion fails). A pressure transducer continuously monitors airway pressure and displays a green light emitting diode (LED) for airway pressures from 0 to 20 cm H_2O, an amber LED for airway pressures from 20 to 30 cm H_2O, and a red LED for airway pressures for greater than 30 cm H_2O. If a prolonged period of "red" pressures is identified, the ventilator delivers a prolonged expiratory phase. The ventilator will not deliver additional breaths if the airway pressure remains high and will alarm. Alternatively, if the ventilator detects a prolonged period of low airway pressures, it will generate an alarm that indicates either a loss of fresh gas supply or circuit disconnect.

Control System and Components

The hardware and software controlling the ventilator were designed by our group specifically for this project. In an effort to make the ventilator highly scalable and affordable, we chose components that were readily available and inexpensive. The ventilator hardware is built on a circuit board with dimensions of about 3 × 3 inches. It is powered by a 120-V wall adapter and has an attached battery backup.

The microcontroller costs under $2 U.S.D at the time of this writing. Gas pressurization and flow are controlled by three 12-V solenoid valves. These valves open and close timed to allow safe and effective ventilation. The timing is adjusted by the user to set respiratory rate, inspiratory time, and sensitivity to spontaneous breathing.

Gas pressure in the airway circuit is measured by a pressure transducer rated for ± 0.5 PSI (24PCEFA6G, Honeywell International, Charlotte, NC) or roughly ± 35 cm H_2O. As previously mentioned, the ventilator continuously illuminates LEDs that correspond to specific pressure levels (0–20, 20–30, and 30+ cm H_2O), providing the user with visual feedback on airway pressures throughout the respiratory cycle without the need for a display screen. The software and hardware have several dedicated safety features, including alarms for sustained high pressure, circuit disconnect or gas supply failure, and electricity supply failure. In addition to generating visual and auditory alarms, sustained high pressure will trigger ventilation to cease and the expiratory valve to remain open until pressure falls below a specific threshold and then ventilation will resume. Although not adjustable by the user, the overpressure threshold is modifiable in the software. The main program simply samples airway pressure, illuminates the corresponding LED for the present airway pressure, and monitors for periods of high or low airway pressures. High- and low-priority interrupts are programmed to handle alarms and timing of solenoid opening and closing throughout the respiratory cycle.

Measurements and Results

Simulation Testing

Human lung simulation was achieved with ALS 5000 (IngMar Medical, Pittsburgh, PA). Compliance and resistance testing was achieved in a manner similar to the process described by Cristiano et al. We compared the Portsmouth Ventilator with both pressure control and volume control with a commercially available ventilator (Dräger Apollo, Dräger, Lubeck, Germany). Three initial trials were completed with the following lung parameters:

- 1) Resistance 12 cm H_2O/L/s and compliance 20 mL/cm H_2O.
- 2) Resistance 12 cm H_2O/L/s and compliance 50 mL/cm H_2O.
- 3) Resistance 15 cm H_2O/L/s and compliance 50 mL/cm H_2O.

After confirmation of acceptable tidal-volume delivery was completed above, ISO Standard 80601-2-12:2020 specifications on volume-control settings were completed for tests 1–7. Additional tests were not completed due to the known tidal-volume limitations of the ventilator (cannot deliver tidal volumes less than 300 mL). A modern commercially available ventilator was similarly tested against the standard for comparison. Waveform data from these simulated tests are displayed in **Figure 2**. Peak inspiratory pressure, plateau pressure, tidal volume delivered, and percent-difference-delivered tidal volume from predicted are summarized in **Table 2**.

	Peak Pressure (cm H_2O)		Plateau Pressure (cm H_2O)		Tidal Volume (mL)		% Difference	
ISO Trial Number	Drager	PV	Drager	PV	Drager	PV	Drager	PV
1	19.8	21.4	15.2	17.2	482	535	−3.60	7.00
2	33.8	46	27	28.2	473	522	−5.40	4.40
3	32.2	28.9	21.9	25.9	490	497	−2.00	−0.60
4	39.9	47.5	35.7	38.5	461	477	−7.80	−4.60
5	30.5	38.1	22.3	26.1	285	325	−5.00	8.33
6	40.7	61.2	31.8	37.5	275	307	−8.33	2.33
7	40.7	63.9	36	45.5	246	275	−18.00	−8.33

PV = Portsmouth Ventilator.

Table 2: ISO Tests with Performance Outcomes with Drager Apollo Compared with Portsmouth Ventilator

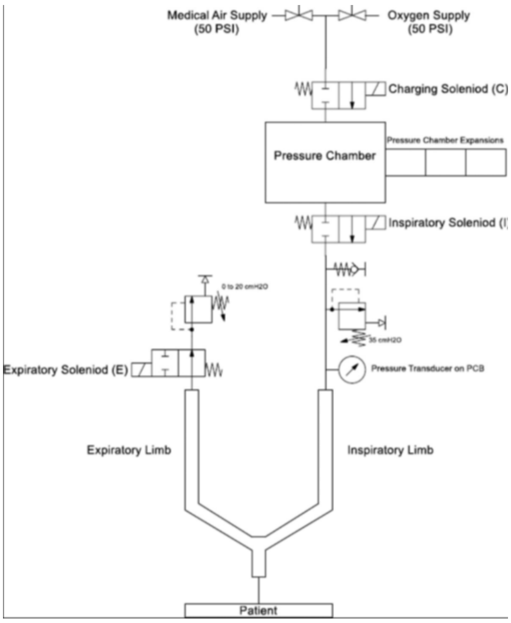


Fig 1.1: Pneumatic system diagram.

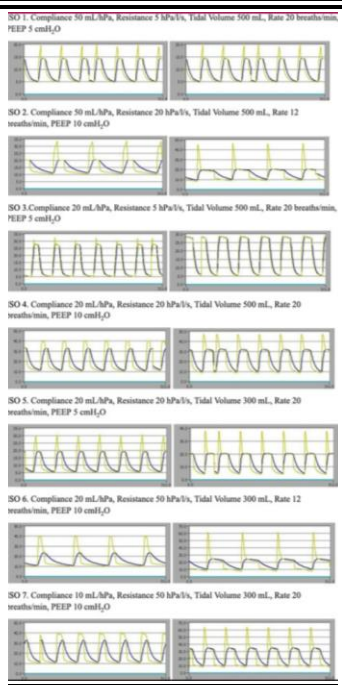


Fig 1.2: Waveform data from Stimulated tests

Comparison of performance Portsmouth Ventilator with the Drager Apollo (Drager, Lubeck, Germany) ventilator with ISO test numbers 1–7. *Yellow waveform* is airway pressure (cm H_2O) and *blue waveform* is tracheal/alveolar pressure. *Left column* is the standard ventilator and the *right column* is the Portsmouth Ventilator. PEEP = positive end-expiratory pressure.

We further tested the device with varying degrees of airway resistance and lung compliance to simulate severe ARDS and chronic obstructive pulmonary disease based on existing literature for lung respiratory parameters. Similarly, we tested extremes of compliance and resistance to validate further the range of pathophysiologic states over which the ventilator can safely operate. This included extremely low compliance with high resistance, extremely high compliance with low resistance, and varying high/low combinations of compliance and resistance. Due to lack of ISO standards for tidal-volume predictability in ventilators that are neither traditional volume control nor pressure control, we set an arbitrary ± 10% range from predicted tidal volume in the trial. This seems clinically appropriate, as it represents less than 1-mL/kg deviation. Because of the large pressure differences in the pulmonary system compared with the pipeline/charging cylinder (20 cm H_2O is equivalent to 0.284 PSI), we assumed the predicted tidal volumes would remain nearly constant over a range of compliance, resistance, and PEEP variables.

To be thorough, other examples of ventilator testing were included. Based on these, we similarly followed the previously described protocol to perform testing at a resistance of 5 cm H_2O/L/s and compliance of 100 mL/cm H_2O, resistance of 20 cm H_2O/L/s and compliance of 30 mL/cm H_2O (ARDS), and resistance of 50 cm H_2O/L/s and compliance of 100 mL/cm H_2O (obstruction). We followed the additional protocol comparing resistance of 5, 10, and 20 cm H_2O with compliances of 30, 70, and 120 mL/cm H_2O. These data are summarized in Supplementary Data File. Testing revealed strongly predictable tidal volumes all within the specified 10% change from baseline. Poorly compliant mechanics were associated with higher plateau pressures and lower tidal volumes, though the ventilator still performed within the standard and was similar to the commercial ventilator.

Other testing describes the impact of adding pressure chamber expansions to the main pressure chamber on tidal volume and the effect of PEEP on the driving pressure needed to generate these tidal volumes. There was a theoretical concern that the tidal volume delivered might increase in a nonlinear manner due to increased airway pressure, but the stepwise volume increases appear to operate in a linear manner across physiologic pressure ranges (R^2 = 0.999). This simulated testing suggests that the proposed mechanism of changing the size of the pressure chamber through the addition or removal of smaller expansion chambers is a reliable and predictable means of modifying the tidal volume that is being delivered to a patient. It also suggests that the Portsmouth Ventilator is able to deliver these tidal volumes at airway pressures that are comparable with other commonly used ventilators.

An additional high-fidelity lung simulator (TestChest, Organis GmbH, Landquart, Switzerland) was used for further simulation testing. ISO standard for volume control ventilators was repeated on the new test lung (tests 1–7). The ventilator was then against the simulated COVID-19 in the lung model. Two models were used: an early model and a late/severe model. The early was characterized by chest wall compliance of 93 mL/hPa, total compliance 52 mL/hPa, and airway resistance 5, whereas the late model had a chest wall compliance of 93 mL/hPa, total compliance 39 mL/hPa, and airway resistance 5. This was similarly tested against the standard ventilator.

In summary, the ventilator performance was similar to existing ventilators across a range of pulmonary mechanics. Changes in PEEP and tidal volumes did not affect predicted tidal volume delivery. Despite changes in airway resistance and compliance, the ventilator was still able to deliver adequate tidal volume breaths. Based on these simulations, the ventilator appeared to be safe for in vivo use.

In Vivo Testing

This study was approved by the Naval Medical Center Portsmouth Institutional Animal Care and Use Committee under protocol number NMCP.2020.0011. A single female 84-kg Yorkshire swine was used for testing. The animal was anesthetized with intramuscular ketamine, acepromazine, and atropine, and placed on 100% FIO2 with 2% isoflurane until intubation. The animal model remained on 100% FIO_2 throughout the study period. Following intubation, the animal was transitioned to a total IV anaesthetic with fentanyl and propofol. The animal was maintained on a standard veterinary mechanical ventilator throughout induction (Hallowell EMC Model 2000, Hallowell EMC, Pittsfield, MA). The animal was paralyzed with rocuronium that was titrated one of four train-of-four twitches. Following induction, the animal was maintained on the standard veterinary Hallowell ventilator for 60 minutes. At t = 60, the animal model was transitioned to the Portsmouth Ventilator, and mechanical ventilation was provided for an additional 120 minutes. We collected arterial blood gas measurements and recorded pH, PO_2, and PCO_2 at t = 0, t = 15, t = 30, t = 45, t = 60, t = 75, t = 90, t = 105, t = 120, t = 135, t = 150, t = 165, and t = 180, where t = 0 corresponds to placement of the arterial line immediately following intubation. Samples from t = 0 to t = 60 reflect standard ventilator function, and all samples beginning at t = 75 reflect the Portsmouth Ventilator. Pulse oximetry and end-tidal CO_2 were recorded at these intervals. Airway pressures were monitored and recorded by an external pressure sensor at 60 Hz, in addition to the sensor in the ventilator. After t = 180, the animal was euthanized per standard veterinary protocols.

Throughout the study period, the respiratory parameters of Portsmouth Ventilator were manipulated by the investigators to provide optimal ventilation and then to test its maximal capabilities through "stress-testing" where the respiratory rate was increased sequentially in an effort to determine the threshold at which the ventilator would no longer provide safe or effective ventilation. These parameters were changed every 15 minutes corresponding with the scheduled arterial blood gas analysis. The respiratory parameters that are reported are correlated with the blood gas analysis that was obtained 15 minutes after the ventilator settings were changed (summarized in **Table 3**). Of note, the animal remained hemodynamically stable throughout the study period.

Time (min)	Respiratory Rate (Beats/min)	Tidal Volume (mL)	Number of Expansions	Positive End-Expiratory Pressure (cm H_2O)	pH	Pao_2 (mm Hg)	$Etco_2$ (mm Hg)	$Paco_2$ (mm Hg)	$Etco_2$ to $Paco_2$ Gradient (mm Hg)	Arterial oxygen saturation (%)
Spontaneous ventilation										
0	20	600	–	0	7.51	355	42	44.2	2.2	99
Conventional ventilator										
15	12	500	–	0	7.49	493	36	48.1	12.1	100
30	18	500	–	0	7.51	481	34	45.5	11.5	100
45	18	500	–	0	7.52	429	33	46.6	13.6	100
60	18	500	–	0	7.5	477	34	47.8	13.8	100
Portsmouth Ventilator										
75	18	–	4	5	7.48	452	42	49.3	7.3	100
90	20	–	5	5	7.48	508	41	50.5	9.5	100
105	20	–	7	5	7.51	392	40	45	5	100
120	20	–	7	5	7.5	461	39	45.6	6.6	99
135	24	–	7	5	7.54	482	35	42.4	7.4	100
150	24	–	7	10	7.56	457	34	41.2	7.2	100
165	24	–	7	10	7.57	365	31	38	7	100
180	30	–	7	10	7.6	428	29	35.5	6.5	100

Table 3: Relevant Ventilator Settings and Measures of Ventilation During Porcine Testing

During this testing, the Portsmouth Ventilator was able to provide adequate ventilation to the 84-kg swine model. The $EtCO_2$-to-$PaCO_2$ gradient found while using the Portsmouth Ventilator was significantly lower than the conventional ventilator. The mean difference of these values was significant based on a two-sided *t* test with *p* value of less than 0.001. This suggests there was an enhanced open lung ventilation strategy when using the Portsmouth Ventilator when compared with the veterinary ventilator. We theorize that this finding is due to the inability of the veterinary ventilator to administer PEEP. The use of PEEP in modern ventilators has been well documented to improve the gradient and is a critical function.

Discussion

The Medicines & Healthcare Products Regulatory Agency in the United Kingdom has released consensus guidelines documenting the minimum acceptable standards that a newly developed ventilator should meet prior to its use on patients who are impacted by the COVID-19 pandemic. This is the only guideline published by a major world government, which details the requirements, and we ensured our ventilator met those requirements. The Portsmouth Ventilator provides a hybrid, pneumatic form of AC ventilation in spontaneously breathing patients. Inspiratory airway pressure in this ventilator is limited to 35 cm H_2O by design as a safety mechanism. The system uses a PEEP valve that is commonly available within hospital systems to provide PEEP while using a self-inflating bag respirator, and notably is the only medical component used in the ventilator.

The inspiratory-to-expiratory (I: E) ratio can be adjusted from 1:1 to greater than 1:5 in the setting of very slow respiratory rates. Likewise, the respiratory rate can be set from 4 to 30 breaths/min. The tidal volume of this ventilator can be adjusted from 300- to 800-mL tidal volumes in 45-mL increments. The Portsmouth Ventilator connects to the wall pipeline air and oxygen supplies using diameter index safety system connectors that are standardized throughout the United States, though these could easily be changed for the local standard connectors wherever this ventilator is needed. By design, this ventilator has a gas reservoir that allows for peak inspiratory flow rates of up to 120 L/min despite the average wall pipeline oxygen supply only providing around 6–10 L/min. The proportioning system is able to provide 50–60% FIO_2 in addition to 90–100% FIO_2 options through mixing of wall pipeline air and oxygen within the gas blender portion of the ventilator. This ventilator also allows for the use of standard connectors to ISO 5356-1:2015.

A commercially available lithium-polymer battery can provide up to 30 minutes of backup function in case of failure of the main electrical system. This system is powered by a U.S. standard 120-V 3 pin plug (this does not meet the U.K. standard, though could be easily converted to allow for stepping down 240V) using a simple direct current (DC) converter. The circuit can be modified to include an on-board voltage regulator to provide 12V DC with other, more commonly available, power sources such as a standard laptop charger.

The ventilator provides an auditory and visual alarm in the event of gas supply failure by detecting whether minimal inspiratory pressures are not achieved for a designated period of time. It also provides auditory and visual alarms in the setting of electricity supply failure should the battery backup be required. If there is a prolonged period of dangerously elevated airway pressures, the ventilator will enter a fail-safe mode. In this mode, an auditory alarm will sound and the ventilator will open the expiratory valve and not resume ventilation until the airway pressures return to a lower level. The ventilator displays the airway pressure in a categorical fashion, with pressures from 0 to 20 cm H_2O powering a green LED, pressures from 20 to 30 cm H_2O powering an amber LED, and pressures above 30 cm H_2O powering a red LED. Although this does not provide the granularity of a digital display, it is simpler and cheaper, and still provides sufficient information to guide ventilator management.

CONCLUSIONS

The current COVID-19 worldwide pandemic may result in limited healthcare resources. Of those, one of the greatest concerns is the risk of limited ventilators. A simple to build and easy to operate ventilator, made with readily available components, could provide a reliable ventilator solution in the case of surge demand. Similarly, because of relatively low cost (< $250), this could potentially provide a ventilator solution in other resource-constricted environments. The Portsmouth Ventilator has limitations compared with modern ventilators; however, we believe it provides a safe, effective, and rapidly scalable alternative ventilation solution.

ACKNOWLEDGEMENTS

We thank the NASA Langley Research Center team (Mary Stringer, Lisa Scott Carnell, and Corey Diebler) for their efforts in establishing a path for future collaboration to obtain a hardware safety review, and manufacturing and path-to-market plans. We also thank the animal care staff at Naval Medical Center Portsmouth, including the attending Veterinarian MAJ Joanna Fishback, DVM, and CAPT John Devlin, MD, who provided material support for in vivo testing. Similarly, we thank the pharmacy staff and, in particular, Tim Gendron, PharmD, and, finally, significant mentorship and guidance from our departmental and hospital leadership, CDR Jason Longwell, MD, and CAPT Marlisa Elrod, MD/PhD.

LITERATURE SURVEY-2

DRDO's Portable Low-Cost Ventilator: "DEVEN" - Transactions of the Indian National Academy of Engineering. A reliable, portable and low-cost ventilator named "DEVEN" is designed and developed by scientists of Dr. APJ Abdul Kalam Missile Complex, RCI

Abstract

A reliable, portable and low-cost ventilator named "DEVEN" is designed and developed by scientists of Dr. APJ Abdul Kalam Missile Complex, RCI, DRDO-Hyderabad. This Ventilator is named as DEVEN—DRDO's Economical Ventilator. DEVEN has features comparable to high-end ventilators and would serve the requirements of a large number of ventilators under the present COVID-19 pandemic situation. Also, DEVEN being a portable ventilator can be used in field application using a portable air compressor and reservoir. Hence, it can be used in an ambulance, any mobile vehicle or for application in any remote/rural area. DEVEN has a micro-controller-operated solenoid valve-based design and is developed by tweaking existing technology being used for hot gas reaction control systems (HRCS), employed in attitude control of exo-atmospheric missiles. HRCS is achieved by actuating solenoid valves through a micro-controller to control the flow of oxidizer as well as fuel. Existing controllers being used for control of electro-mechanical actuators are tweaked for control of the above-mentioned solenoid valves for inspiratory as well as expiratory lines of DEVEN.

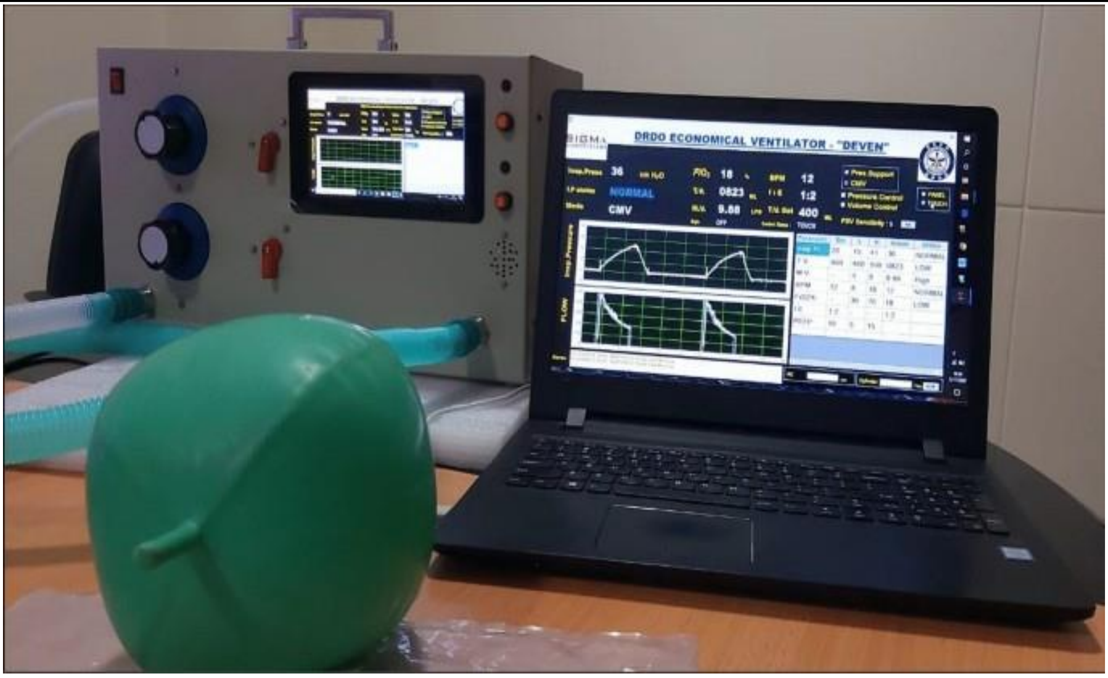


Fig 1.3: Graphic abstract

Technology

Developed a low-cost portable ventilator with precise measurement and control/adjustment of important patient parameters such as inspiratory pressure, respiration (breathe) rate, inhaling–exhaling (I:E) ratio, tidal volume, and percentage oxygen (FiO_2). All the above-mentioned parameters are controlled using manual valves/touch screen and are displayed using serial communication link onto a touch-screen LED display. Once displayed, the doctor/hospital attendants can vary these parameters depending upon the patient's breathing requirement. Different categories of patients have different respiration rates and hence require different inhalation and exhalation of air as well as oxygen. All the parameters (except FiO_2) are controlled on the touch screen which in turn controls the opening and closing times of the solenoid valves. FiO_2 is controlled manually by suitably positioning the knob of a three-way valve. Air and oxygen enter through this valve at two inlets and a mixture of both air and oxygen comes out from the outlet of this valve and this mixture of both gases is subjected to the patient through the inspiratory solenoid valve.

Methodology for Design and Development of DEVEN

The following methodology was adopted for design and development:

1. (i)

Essential technical features of low-cost ventilators for COVID-19 situation, put forth by empowered technical committee of DRDO, were studied.

2. (ii)

Requirements of medicine and healthcare products regulatory agency (MHRA)-UK for rapidly manufactured ventilator system (RMVS 2020) were also studied.

3. (iii)

Existing open-source designs for low-cost ventilators were studied which were mostly AMBU bag-based designs (Emergency Ventilation Alternative System 2020; Read 2020) and have inherent disadvantages.

4. (iv)

A new design was conceptualized using solenoid valves actuated by a micro-controller. It was ensured that all commercially-off-the-shelf (COTS) available parts having low cost are used.

5. (v)

A functional prototype was developed using available parts and functioning of the same was demonstrated to doctors from various hospitals in Hyderabad.

6. (vi)

Detailed design review was conducted by a committee with Director DEBEL-DRDO as chairperson, many senior scientists from DRDO and two doctors as members. This Committee was satisfied with the simplicity of the design of DEVEN and its features such as control of parameters, display and alarms.

7. (vii)

Order has been placed for the realization of ten DEVEN ventilators on industry partner. These units would be subjected to further testing, evaluation and approvals.

8. (viii)

Endurance testing for 5 days and battery backup testing are in progress.

9. (ix)

Future tasks planned are as follows:

10. (a)

Testing using a calibrator set-up for confirmation of all patient parameters and their displayed values/alarms.

11. (b)

Demonstration to empowered technical committee of Central Government for regulatory approval and clearance for mass production.

Working Principle of DEVEN

Overall schematic and electrical block diagram are shown in Figs. 1 and 2, respectively.

Fig 1: Overall schematic diagram of DEVEN

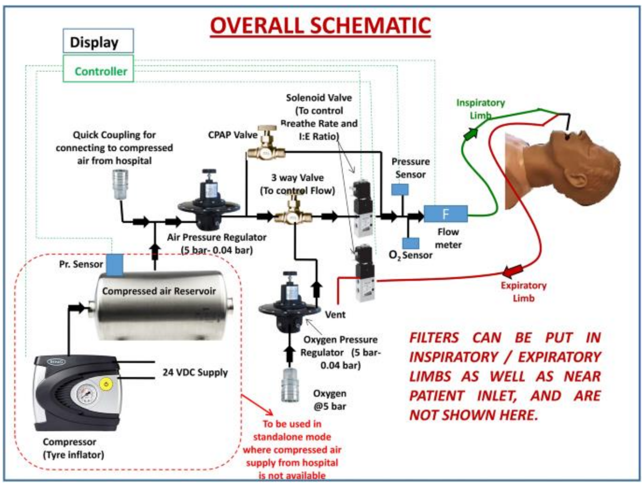
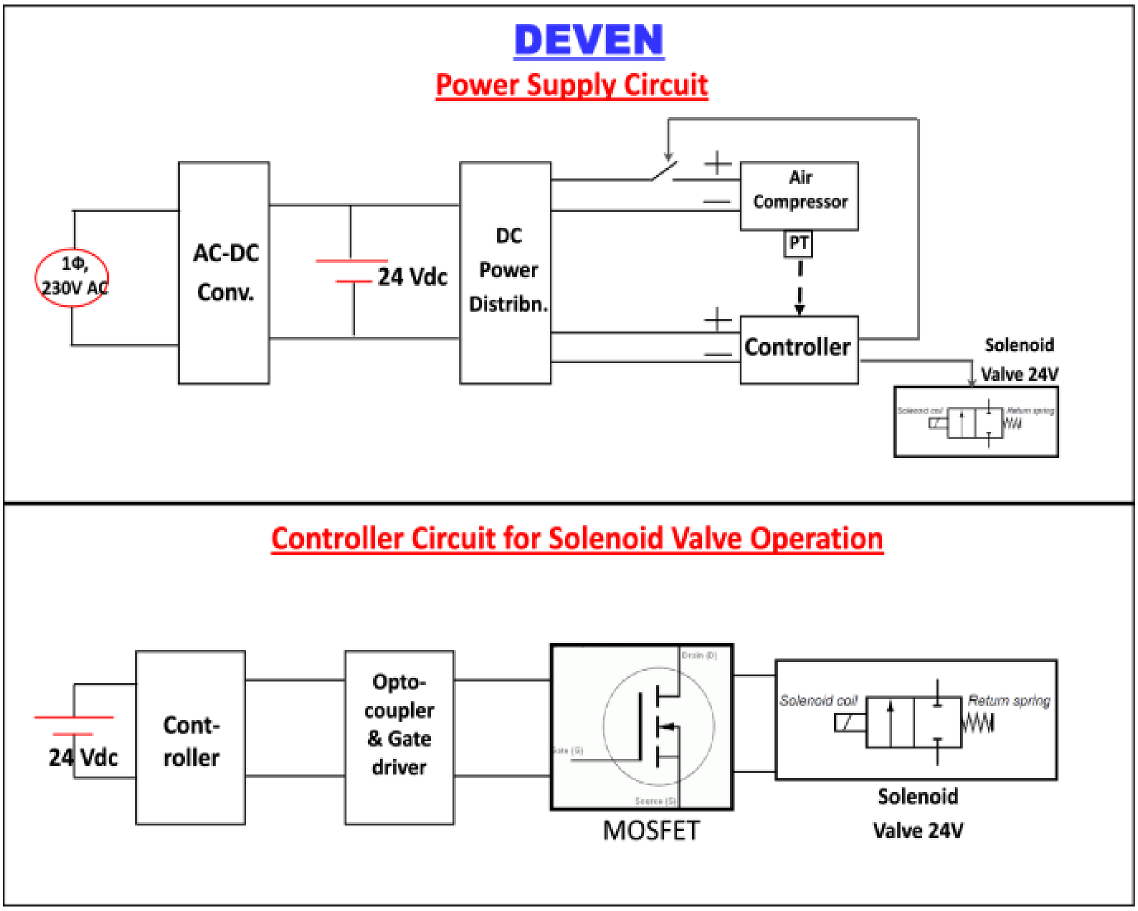


Fig. 2: Electrical block diagram of DEVEN



Working Principle of DEVEN can be Explained as Follows

DEVEN ventilator system has two modes of operation (as shown in Fig. 1):

1. (a)

Hospital mode.

2. (b)

Stand-alone mode.

In hospital mode, DEVEN draws compressed air and oxygen, both available at 5 bar pressure from centralized hospital compressed air and oxygen supply lines. In standalone mode, compressed air at 5 bar pressure is supplied from compressor and air reservoir, and oxygen is supplied from an oxygen cylinder, through a pressure regulator. This pressure regulator reduces the pressure of oxygen (from cylinder) to 5 bar.

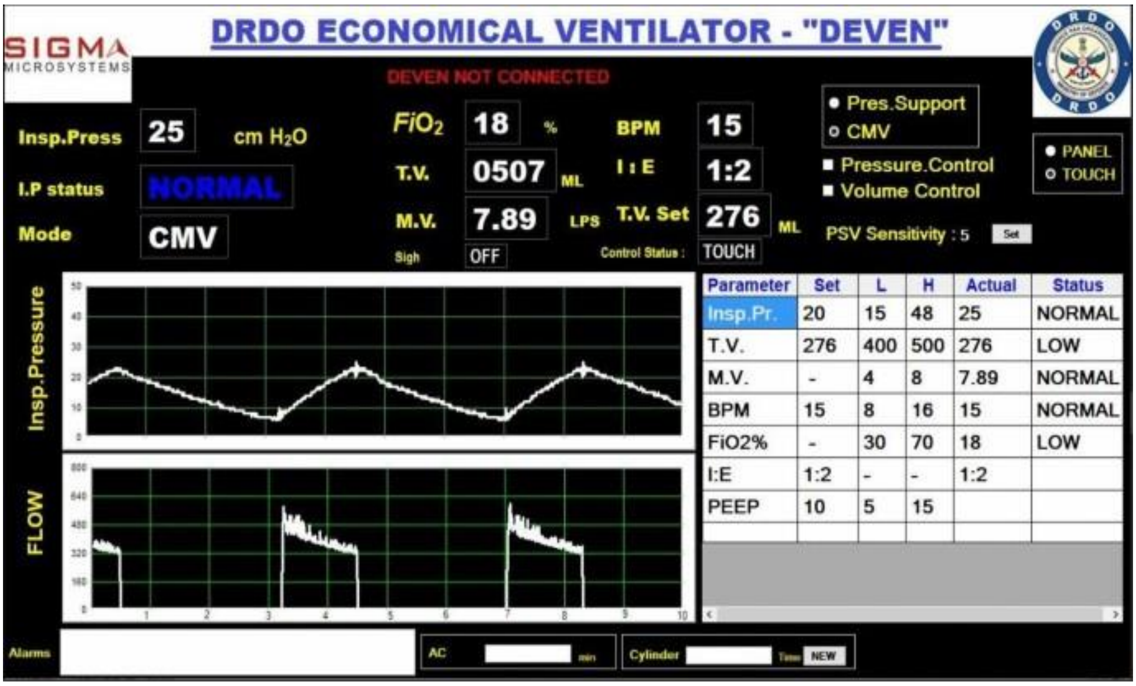
Further Working is Same for Both the Modes of Operation and is Explained as Follows

Compressed air and oxygen supplies at 5 bar pressure are available at the inlet of low-pressure air and oxygen regulators built into the ventilator (Al Husseini 2010). The outlet pressure of these pressure regulators is adjustable between 10 and 50 cm H_2O. There is a three-way (01 no.) or two-way (02 nos.) manual control ball valve(s) at the outlet of these two pressure regulators. Mixing of air and oxygen takes place in these valves. Also, these valves are used to control the flow of air/oxygen being subjected to the patient at a pressure between 10 and 50 cm WC (adjustable). 100% air, 100% oxygen or a mixture of the two can be given to the patient using this ball valve(s).

Mixture of air/oxygen is fed to the inspiratory limb of the patient through solenoid valve. Opening and closing of this solenoid valve will start/stop the inspiration. Frequency of operation of solenoid valve, i.e. its ON/OFF time, is controlled by a micro-controller depending upon the requirement of respiration rate (BPM), I:E ratio and tidal volume.There are three sensors in the inspiratory line for measuring FiO_2, inspiratory pressure and flow rate of air/oxygen being delivered to the patient. Values of FiO_2, inspiratory pressure and flow rate of air/oxygen being delivered to the patient are read by the controller and displayed on to an LED display. Also, the limits of these parameters can be set from the LED touchscreen using touch and alarms are programmed in case the measured values cross these limits in case of any abnormality. If necessary, a non-return valve (NRV) can be provided in the inspiratory line to avoid any backflow from the patient.

During exhalation, solenoid valve in the inspiratory line will close and solenoid valve in the expiratory line will open. Hence, flow can take place from the patient through the expiratory line. If necessary, a non-return (NRV) valve can be provided in expiratory line to avoid any backflow to the patient. Frequency of operation of these two solenoid valves will decide the BPM, I:E ratio, tidal volume, PEEP, etc. Display of all these parameters on an LCD screen is shown in Fig. 3.

Fig. 3: Display screen of DEVEN



Safety Features

Since a ventilator works as a life support system, various safety features are required. Safety features provided in DEVEN are as follows.

Alarms (Audio Visual)

1. (i)

High- and low-level alarms for inspiration pressure, tidal volume, BPM, I:E ratio, FiO_2 and PEEP.

2. (ii)

'Patient disconnected' alarm in case the inspiratory or expiratory limbs are disconnected accidentally.

3. (iii)

'Ventilation abnormal' alarm in case the breathing pulse is not detected by the ventilator system.

4. (iv)

‘Power failure’ alarm.

These alarms will alert the hospital staff in case any abnormality is detected. Alarm history of the last 200 alarms is provided which can be retrieved on the screen if desired by the hospital staff.

Interlock During Switching Off

There is an interlock provided in case the ventilator gets accidentally switched off. This is ensured with an over-rider command on the touchscreen display to accept switch-off. Hence, double confirmation is achieved for switching off the ventilator, if someone wants to switch off.

Power Back-Up

Redundancy for power failure is provided with the help of a UPS, which can keep the ventilator operational for 2 hours in case of main power failure.

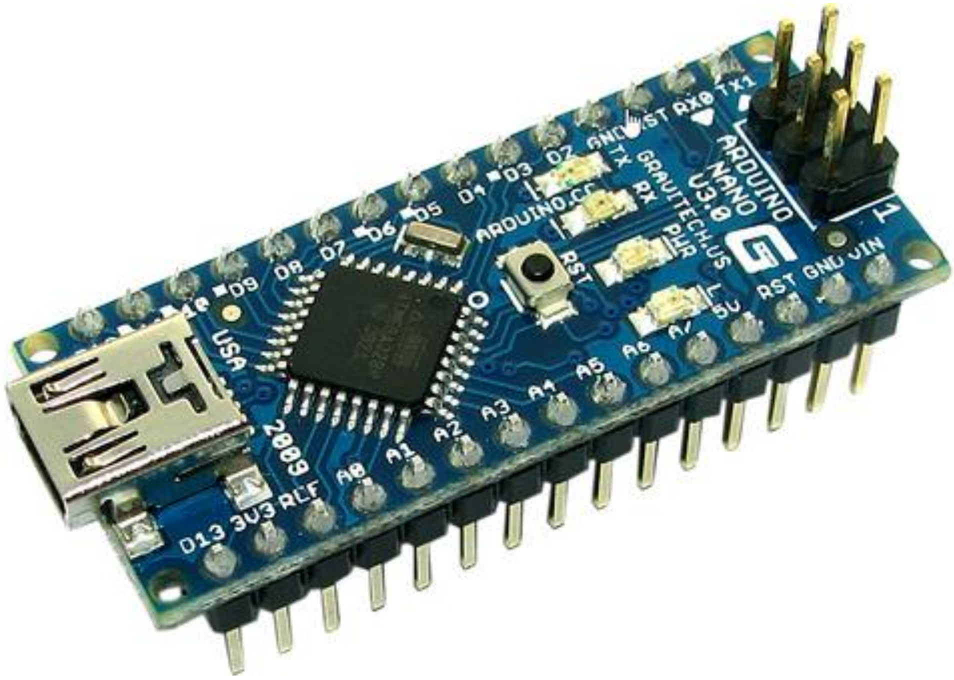
Results

DEVEN has been successfully developed and functioning has been demonstrated to various hospitals. All the parameters were checked for display and all the alarms have been tested. Endurance testing, i.e. continuous working of DEVEN for 5 days and battery backup testing is in progress. Subsequently, DEVEN will be tested using a calibrator unit to validate the values of various parameters read by sensors and displayed on the screen.

CHAPTER 3

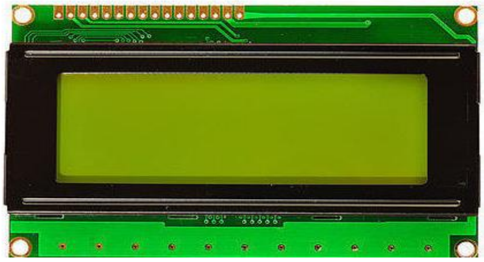
COMPONENTS USED

1. Arduino Nano



The Arduino Nano is Arduino's classic breadboard friendly designed board with the smallest dimensions. The Arduino Nano comes with pin headers that allow for an easy attachment onto a breadboard and features a Mini-B USB connector. The classic Nano is the oldest member of the Arduino Nano family boards. It is similar to the Arduino Duemilanove but made for the use of a breadboard and has no dedicated power jack. Successors of the classic Nano are for example the Nano 33 IoT featuring a WIFI module or the Nano 33 BLE Sense featuring Bluetooth® Low Energy and several environment sensors. It is based on the ATmega328 (Arduino Nano 3.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.

2. 20*4 LCD



A 20x4 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A **20x4 LCD** means it can display 20 characters per line and there are 4 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. This is standard HD44780 controller LCD.4

3. MAX 30100 PULSE OXIMETER

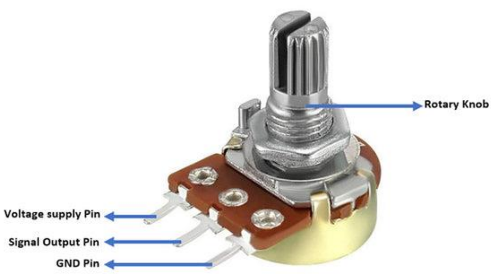
The MAX30100 pulse oximeter and heart rate sensor is an I2C-based low-power plug-and-play biometric sensor. It can be used by students, hobbyists, engineers, manufacturers, and game & mobile developers who want to incorporate live heart-rate data into their projects. The module features the MAX30100 – a modern, integrated pulse oximeter and heart rate sensor IC, from Analog Devices. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry (SpO2) and heart rate (HR) signals.

On the right, the MAX30100 has two LEDs – a RED and an IR LED. And on the left is a very sensitive photodetector. The idea is that you shine a single LED at a time, detecting the amount of light shining back at the detector, and, based on the signature, you can measure blood oxygen level and heart rate. The MAX30100 chip requires two different supply voltages: 1.8V for the IC and 3.3V for the RED and IR LEDs. So the module comes with 3.3V and 1.8V regulators. This allows you to connect the module to any microcontroller with 5V, 3.3V, even 1.8V level I/O.

4. MG995 Plastic Gear High Speed Servo Motor

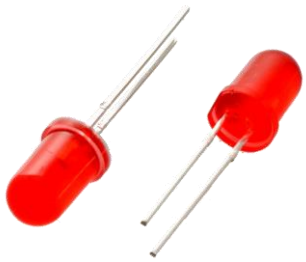
The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec. This high-speed standard servo can rotate 360 degrees. You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast.

5. 10K Ohm Potentiometer



Potentiometers are very useful in changing the electrical parameters of a system. It is a single turn **10k Potentiometer** with a rotating knob. These potentiometers are also commonly called as a **rotary potentiometer** or just **POT** in short. These three-terminal devices can be used to vary the resistance between 0 to 10k ohms by simply rotating the knob. A **potentiometer knob** can also be used along with this POT for aesthetic purposes.

6. LED



Led Emitting Diode - LED is a particular type of diode that converts electrical energy into light. LED illuminates when electricity passes through it. Like all diodes, electricity only flows in one direction through these components. The anode, which typically connects to power is usually longer leg and the cathode is the shorter leg.

7. Push Button Switch

Tactile Push Button Switch is widely used as a standard input "buttons" on electronic projects. These work best when you mount it on PCB but can also be used on a solderless breadboard for temporary connections in prototypes. The pins are normally open (disconnected) and when the button is pressed, they are momentarily closed and complete the circuit.

8. Ambu bag

They are instruments used to provide oxygen during intermittent positive pressure respiration (IPPR) via an endotracheal tube or a facemask. They are used in emergencies when somebody is facing breathing difficulties to provide artificial ventilation. It is a compressible, self-inflating, non-rebreathing silicon bag, which has an inlet through which air and additional O2 is supplied and an outlet through this can be transferred to the patient. The gas enters in the self-expanding bag through one way valve which restricts the flow back from the inlet. When the bag is compressed, the air is pushed forward through the mask in the pharynx or throat which in turn leads to wind pipe and then in the lungs, hence assisting in artificial ventilation. There are other valves which prevent rebreathing of the expired air and excessive pressure from developing. Some of these self-inflating bags come with pressure restrictor or manometer tube connector or pressure gauge connector. The manometer connector (or the pressure gauge connector) can be used to connect manometer tube (or pressure gauge) to monitor the airway pressure. It is strongly recommended to have one of these safety features in self-inflating bags when they are used in paediatric cases.

CHAPTER 4

WORKING AND METHODOLOGY

The development process of a medical device as an embedded real-time system can be divided into the main following steps:

1.System design
2.Schematic development
3.Fabrication and assembly
4.Software development
5.Testing

Each of the above steps undergoes numerous iterations, starting with a concept passing the basic and detailed engineering stages, and ending with a finished product. This study of ventilator systems is based on fundamental works. In addition to the technical difficulties with the development of an embedded real-time system, there are also a significant number of details associated with the fabrication of parts that are used in contact with the patient. The developed system has three control inputs for the variables: tidal volume (V_T), breathing rate per minute (BPM), and inspiratory-to-expiratory ratio (I/E). BPM and I/E are controlled by rotary potentiometers, and BPM is controlled with a rotary encoder. Having a rotary encoder with an additional button may allow developers to upgrade the system in the future (for example, add a menu to select another mode).

The self-inflating bag compression process . At the beginning of the operation, the pusher reaches the home position by hitting the limit switch. From this point, the tidal volume can be adjusted by the amplitude of the movement of the pusher (ΔL), and the breathing rate can be adjusted by a pusher frequency. A breathing control diagram is presented. According to the stepper motors datasheets, both the widely used NEMA-17 and NEMA-23 stepper motors have 1.8 degrees per step, which would give $N = 365/1.8 \approx 203$ steps per one revolution of the shaft. With specified micro-stepping multiplier, $k = 2\ldots16$, it is possible to increase the number of steps per one revolution and provide a more smooth and stable rotation of the motor shaft.

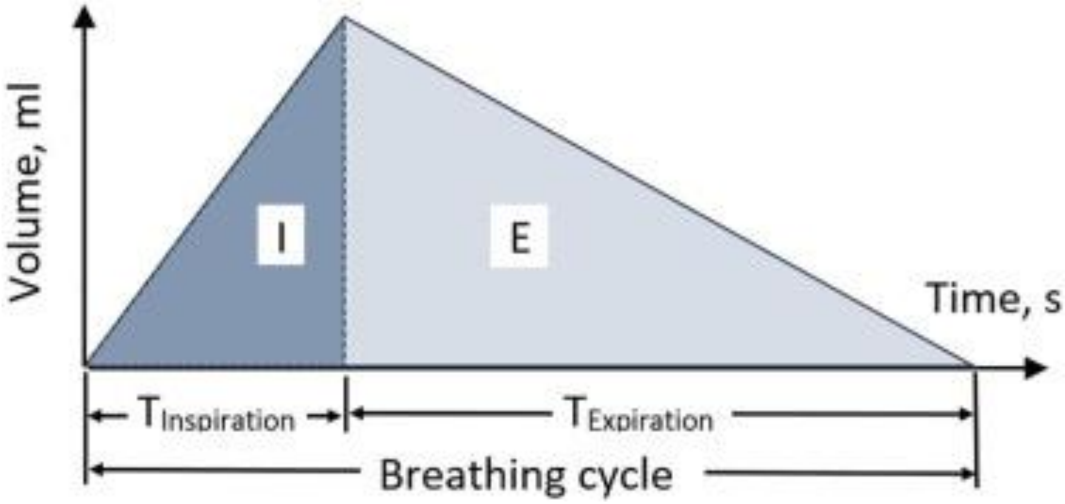


Fig 4.1 Breathing control diagram: the tidal volume depends on the length of extension of the pusher, and the timings for the inspiratory and expiratory phases – are functions of stepper motor delays between its successive steps.

The thrust of the motor depends on the motor torque and the diameter of the gear according to the following equation: F = 2 T/R, where R – is the gear radius and T – is the motor torque. Therefore, by varying the motor current and the size of the gear, it was experimentally found that the herringbone gear (double helical gear) with a diameter of 15 mm will provide reasonable thrust and consistency of contacts between the gear teeth. As V_T, BPM, and I/E are functions of the number of steps and the speed of the stepper motor. To provide the desired breathing parameters, the number of motor steps should be calculated as follows:

(1)$n=\Delta L \cdot N/\pi \cdot D$

where D is the gear diameter in millimetres, ΔL is the desired pusher length in millimetres, and N is the number of steps per one full revolution. At the same time, N = k · 365/1.8 steps, where k is the micro-stepping multiplier (usually k varies from 2 to 16). A greater number of steps per revolution of the motor shaft allows smooth rotation and prevents unwanted vibration of the entire apparatus. It is worth noting, however, that the use of micro-stepping higher k values reduces the overall torque of the motor. Thus, a balance was experimentally found between the number of motor steps and the permissible vibration of the bag support system with a micro-stepping coefficient of 4, which corresponds to ~800 steps per single revolution of the shaft. The volume of air or gas mixture provided by the self-inflating bag is largely due to the shape and size of the pusher. The experiments with three pushers with a total area of 14, 42, and 74 square centimetres revealed linear relationships between the volume of air supplied to the lungs and the pusher travel distance. The linear dependency between the pusher travel distance and provided tidal volume equals to $\Delta L = (83 + V_T)/11.2$ mm. The control system is based on the Arduino controller and a servo motor setup. The Arduino Nano board was chosen as a controller due to low relative expense while having sufficient digital and analog pins.

A significant number of medical software development standards contain information and requirements regarding software design, validation, and certification. However, in the global pandemic, meeting all requirements can be difficult. The main guidelines for emergency ventilation systems is the use of real-time operating systems and a serial peripheral interface for connecting sensing devices . The use of an open-source real-time operating system library for Arduino considerably expands the possibilities of the controller. A real-time operating system provides essential functions to software tasks, such as scheduling, dispatching, inter-task communication, and synchronization. The software system architecture. There are three parallel tasks with equal priorities communicating with the two instances of the patient and nurse classes, which provide scalability (there may be more “patients” and “nurses”, as well as threads with other functions) and possibility of transition to another hardware background since it supports most popular processors and microcontrollers

AGE	Heart Rate	Respiratiory Rate	BP/(SBP/DBP)
Premature	120-170	40-70	75-55/45-35
0-3 months	100-150	35-55	85-65/55-45
3-6 months	90-120	30-45	90-70/65-50
6-12 months	80-120	25-40	100-80/65-55
1-3 years	70-120	20-30	105-90/70-55
3-6 years	65-120	20-25	110-95/75-60
6-12 years	60-95	14-22	120-100/75-60
12+ years	55-85	14-18	135-110/85-65

PROGRAM

WIFI OXIMETER

```
#define BLYNK_TEMPLATE_ID "TMPLOkAGdxpc"
#define BLYNK_DEVICE_NAME "Quickstart Template"
#define BLYNK_AUTH_TOKEN "kyY6MRua0qYzTU5rGkvWnMk6cKh5ob8y"



#include <Wire.h>
#include <Blynk.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include "MAX30105.h"
#include "spo2_algorithm.h"
#include "heartRate.h"




MAX30105 particleSensor;

#define MAX_BRIGHTNESS 255

uint32_t irBuffer[100]; //infrared LED sensor data
uint32_t redBuffer[100];  //red LED sensor data

#define REPORTING_PERIOD_MS 1000 // frequency of updates sent to blynk app in
ms

char auth[] = "kyY6MRua0qYzTU5rGkvWnMk6cKh5ob8y";            // You should get
Auth Token in the Blynk App.
char ssid[] = "WiFiOximeter";                         // Your WiFi credentials.
char pass[] = "1234567890";
```

```
uint32_t tsLastReport = 0;  //stores the time the last update was sent to the blynk app

int32_t bufferLength; //data length
int32_t spo2; //SPO2 value
int8_t validSPO2; //indicator to show if the SPO2 calculation is valid
int32_t heartRate; //heart rate value calcualated as per Maxim's algorithm
int8_t validHeartRate; //indicator to show if the heart rate calculation is valid

byte pulseLED = 2; //onboard led on esp32 nodemcu
byte readLED = 19; //Blinks with each data read

long lastBeat = 0; //Time at which the last beat occurred

float beatsPerMinute; //stores the BPM as per custom algorithm
int beatAvg = 0, sp02Avg = 0; //stores the average BPM and SPO2
float ledBlinkFreq; //stores the frequency to blink the pulseLED



void setup()
{
  //ledcSetup(0, 0, 8); // PWM Channel = 0, Initial PWM Frequency = 0Hz, Resolution =
8 bits
  //ledcAttachPin(pulseLED, 0); //attach pulseLED pin to PWM Channel 0
  //ledcWrite(0, 255); //set PWM Channel Duty Cycle to 255
  Serial.begin(115200);
  Serial.print("Start");
  Blynk.begin(auth, ssid, pass);



  Serial.print("Initializing Pulse Oximeter..");

    // Initialize sensor
```

```
if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
{
  Serial.println(F("MAX30105 was not found. Please check wiring/power."));
  while (1);
}



/*The following parameters should be tuned to get the best readings for IR and RED LED.
 *The perfect values varies depending on your power consumption required, accuracy, ambient light, sensor mounting, etc.
 *Refer Maxim App Notes to understand how to change these values
 *I got the best readings with these values for my setup. Change after going through the app notes.
 */
byte ledBrightness = 50; //Options: 0=Off to 255=50mA
byte sampleAverage = 1; //Options: 1, 2, 4, 8, 16, 32
byte ledMode = 2; //Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
byte sampleRate = 100; //Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
int pulseWidth = 69; //Options: 69, 118, 215, 411
int adcRange = 4096; //Options: 2048, 4096, 8192, 16384

particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth, adcRange); //Configure sensor with these settings
}

void loop()
{
  bufferLength = 100; //buffer length of 100 stores 4 seconds of samples running at 25sps

  //read the first 100 samples, and determine the signal range
  for (byte i = 0 ; i < bufferLength ; i++)
  {
```

```
    while (particleSensor.available() == false) //do we have new data?
      particleSensor.check(); //Check the sensor for new data

    redBuffer[i] = particleSensor.getIR();
    irBuffer[i] = particleSensor.getRed();
    particleSensor.nextSample(); //We're finished with this sample so move to next
sample

    Serial.print(F("red: "));
    Serial.print(redBuffer[i], DEC);
    Serial.print(F("\t ir: "));
    Serial.println(irBuffer[i], DEC);
  }

  //calculate heart rate and SpO2 after first 100 samples (first 4 seconds of samples)
  maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2,
&validSPO2, &heartRate, &validHeartRate);

  //Continuously taking samples from MAX30102.  Heart rate and SpO2 are calculated
every 1 second
  while (1)
  {
    Blynk.run();
    //dumping the first 25 sets of samples in the memory and shift the last 75 sets of
samples to the top
    for (byte i = 25; i < 100; i++)
    {
      redBuffer[i - 25] = redBuffer[i];
      irBuffer[i - 25] = irBuffer[i];
    }

    //take 25 sets of samples before calculating the heart rate.
    for (byte i = 75; i < 100; i++)
    {
```

```
while (particleSensor.available() == false) //do we have new data?
  particleSensor.check(); //Check the sensor for new data

digitalWrite(readLED, !digitalRead(readLED)); //Blink onboard LED with every data read

redBuffer[i] = particleSensor.getRed();
irBuffer[i] = particleSensor.getIR();
particleSensor.nextSample(); //We're finished with this sample so move to next sample

//send samples and calculation result to terminal program through UART
//Uncomment these statements to view the raw data during calibration of sensor.
//When uncommented, beatsPerMinute will be slightly off.
/*Serial.print(F("red: "));
Serial.print(redBuffer[i], DEC);
Serial.print(F("\t ir: "));
Serial.print(irBuffer[i], DEC);
Serial.print(F("\t HR="));
Serial.print(heartRate, DEC);
Serial.print(F("\t"));
Serial.print(beatAvg, DEC);

Serial.print(F("\t HRvalid="));
Serial.print(validHeartRate, DEC);

Serial.print(F("\t SPO2="));
Serial.print(spo2, DEC);

Serial.print(F("\t SPO2Valid="));
Serial.println(validSPO2, DEC);*/

long irValue = irBuffer[i];
```

```
    //Calculate BPM independent of Maxim Algorithm.
    if (checkForBeat(irValue) == true)
    {
      //We sensed a beat!
      long delta = millis() - lastBeat;
      lastBeat = millis();

      beatsPerMinute = 60 / (delta / 1000.0);
      beatAvg = (beatAvg+beatsPerMinute)/2;

      if(beatAvg != 0)
        ledBlinkFreq = (float)(60.0/beatAvg);
      else
        ledBlinkFreq = 0;
      ledcWriteTone(0, ledBlinkFreq);
    }
    if(millis() - lastBeat > 10000)
    {
      beatsPerMinute = 0;
      beatAvg = (beatAvg+beatsPerMinute)/2;

      //if(beatAvg != 0)
        //ledBlinkFreq = (float)(60.0/beatAvg);
      //else
        //ledBlinkFreq = 0;
      //ledcWriteTone(0, ledBlinkFreq);
    }
  }

  //After gathering 25 new samples recalculate HR and SP02
  maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2,
&validSPO2, &heartRate, &validHeartRate);

  Serial.print(beatAvg, DEC);
```

```
    Serial.print(F("\t HRvalid="));
    Serial.print(validHeartRate, DEC);

    Serial.print(F("\t SPO2="));
    Serial.print( sp02Avg , DEC);

    Serial.print(F("\t SPO2Valid="));
    Serial.println(validSPO2, DEC);

    //Calculates average SPO2 to display smooth transitions on Blynk App
    if(validSPO2 == 1 && spo2 < 100 && spo2 > 0)
    {
      sp02Avg = (sp02Avg+spo2)/2;
    }
    else
    {
      spo2 = 0;
      sp02Avg = (sp02Avg+spo2)/2;;
    }

     //Send Data to Blynk App at regular intervals
    if (millis() - tsLastReport > REPORTING_PERIOD_MS)
    {
      Blynk.virtualWrite(V3, beatAvg);
      Blynk.virtualWrite(V4, sp02Avg);

      tsLastReport = millis();
    }
  }
}
```

ARDUINO NANO

```
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

//--------------------------

LiquidCrystal_I2C lcd(0x27, 20, 4);

//--------------------------

const int FASTEST_FREQUENCY = 45; // in "breaths per minute"
const int SLOWEST_FREQUENCY = 1; // in "breaths per minute"
const int SERVO_START = 0; // angle in degrees. Will correspond to "0" on the 0-500 volume range
const int SERVO_MAX = 180; // angle in degrees. Will correspond to "500" on the 0-500 volume range

Servo myservo;
int servopin (8);

char buffer[16]; //buffer for sprintf for LCD display.
unsigned long loopTimer = 0;
unsigned long breathStart = 0;
int loopPeriod = 50; // loop period in milliseconds; eg 100 = 10Hz

//--------------------------------------

void setup()
{
  Serial.begin(115200);

  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
```

```
  lcd.init();                      // initialize the lcd
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(4,0);
  lcd.print("Ventilator On");

  myservo.attach (servopin);
  myservo.write (SERVO_START);
  delay (1000);

  loopTimer = millis ();
  breathStart = loopTimer;
}

void loop()
{
  unsigned long currentTime = millis();

  // this loop executes once every loopPeriod milliseconds
  if ( (currentTime - loopTimer) >= loopPeriod ) {
    loopTimer = currentTime;

    // read in potentiometer values
    int pot0 = analogRead (A0);
    int pot1 = analogRead (A1);

    // calculate values used for display
    int breathFrequency = map (pot0, 0, 1023, SLOWEST_FREQUENCY,
FASTEST_FREQUENCY);
    int breathDepthDisplayValue = map (pot1, 0, 1023, 0, 500); // 0..500 scale used for
display only.

    // write the display values
```

```
lcd.setCursor(0, 1);
sprintf(buffer,"BPM:%2d  ",breathFrequency);
lcd.print(buffer);

lcd.setCursor(0, 2);
sprintf(buffer,"VOL:%3d ",breathDepthDisplayValue);
lcd.print(buffer);



//calculate values used for setting values
unsigned long breathLength = 60000 / breathFrequency; // length = period in milliseconds = 60000 / frequency
//unsigned long breathLength = map (pot0, 0, 1023, 60000 / SLOWEST_FREQUENCY, 60000 / FASTEST_FREQUENCY);
unsigned long breathDepth = map (pot1, 0, 1023, SERVO_START, SERVO_MAX); // breathDepth measured in degrees for the servo.

//Check if we've started a new breath
if ( (currentTime - breathStart) >= breathLength) {
  breathStart = currentTime;
}

//Work out how far we are through the current breath, then set the servo target accordingly.
float howFarThroughBreath = (currentTime - breathStart) / (float) breathLength; //0...1 scale for how far through the whole breath cycle we currently are,
float targetDepthFactor; // 0...1 scale for how deep the "diaphragm" should currently be; equals 1.0 when howFarThroughBreath=0.5
if (howFarThroughBreath >= 0.5) {
  //we're breathing out
  targetDepthFactor = (1.0 - howFarThroughBreath) * 2.0;
} else {
  //we're breathing in
  targetDepthFactor = howFarThroughBreath * 2.0;
```

```
    }

    int servoTarget = (int) (targetDepthFactor * (breathDepth - SERVO_START) + 0.5);
    myservo.write (servoTarget);
    Serial.println(servoTarget);
  }
}
```

Circuit Diagram

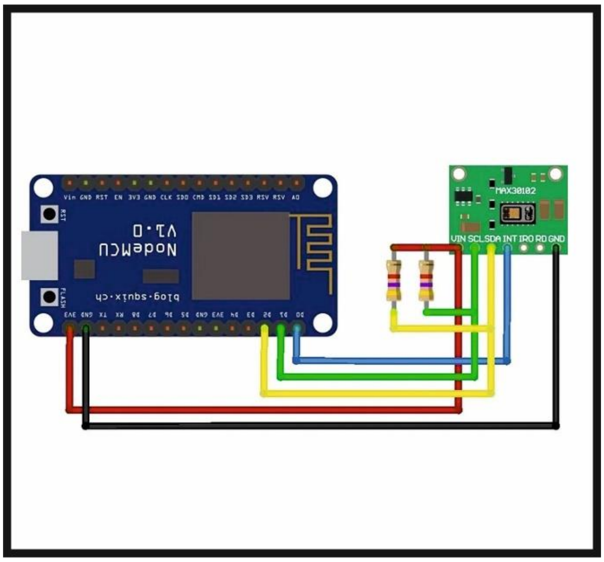


Fig 4.2: SpO2 Monitoring

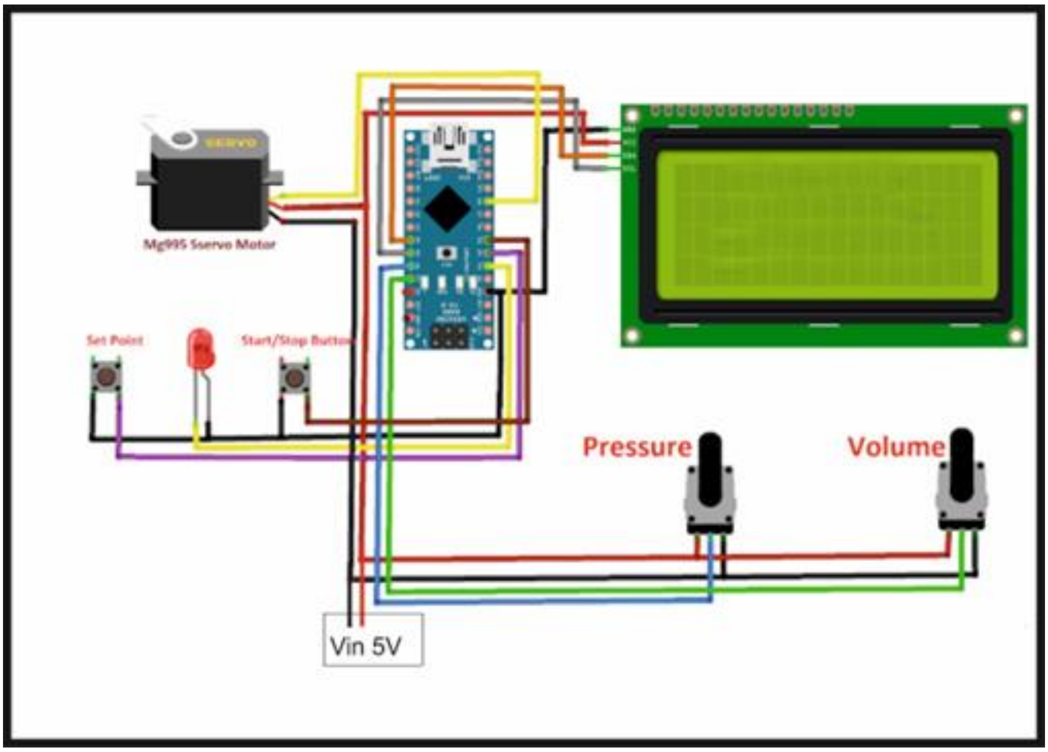


Fig 4.3: Circuit of ventilator

CHAPTER 5

RESULT AND MODEL IMAGES

As we see the servo motor applies pressure on the ambu bag the oxygen id pumped. The amount of pressure and volume to be applied can be monitored by the potentio meter. The values of pressure and volume is displayed on the LED display. So based on the we can change the values depending on the patient condition.

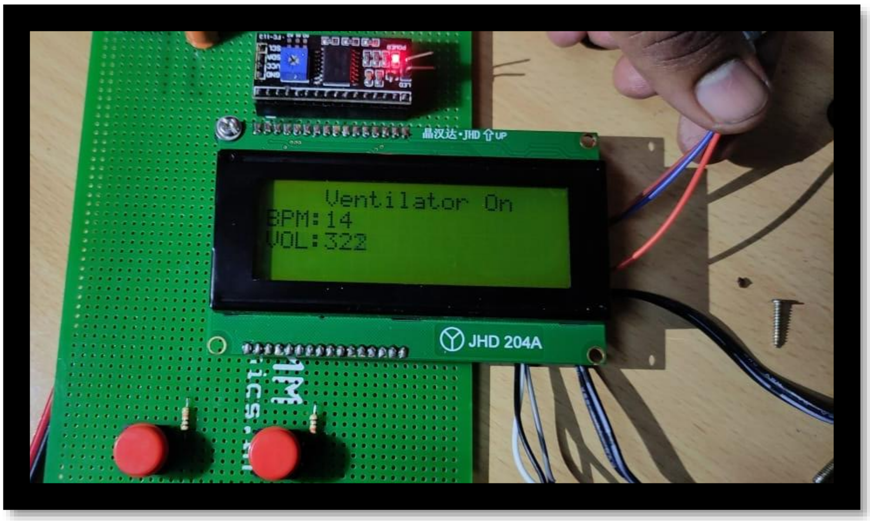


Fig 5.1: LCD DISPLAY

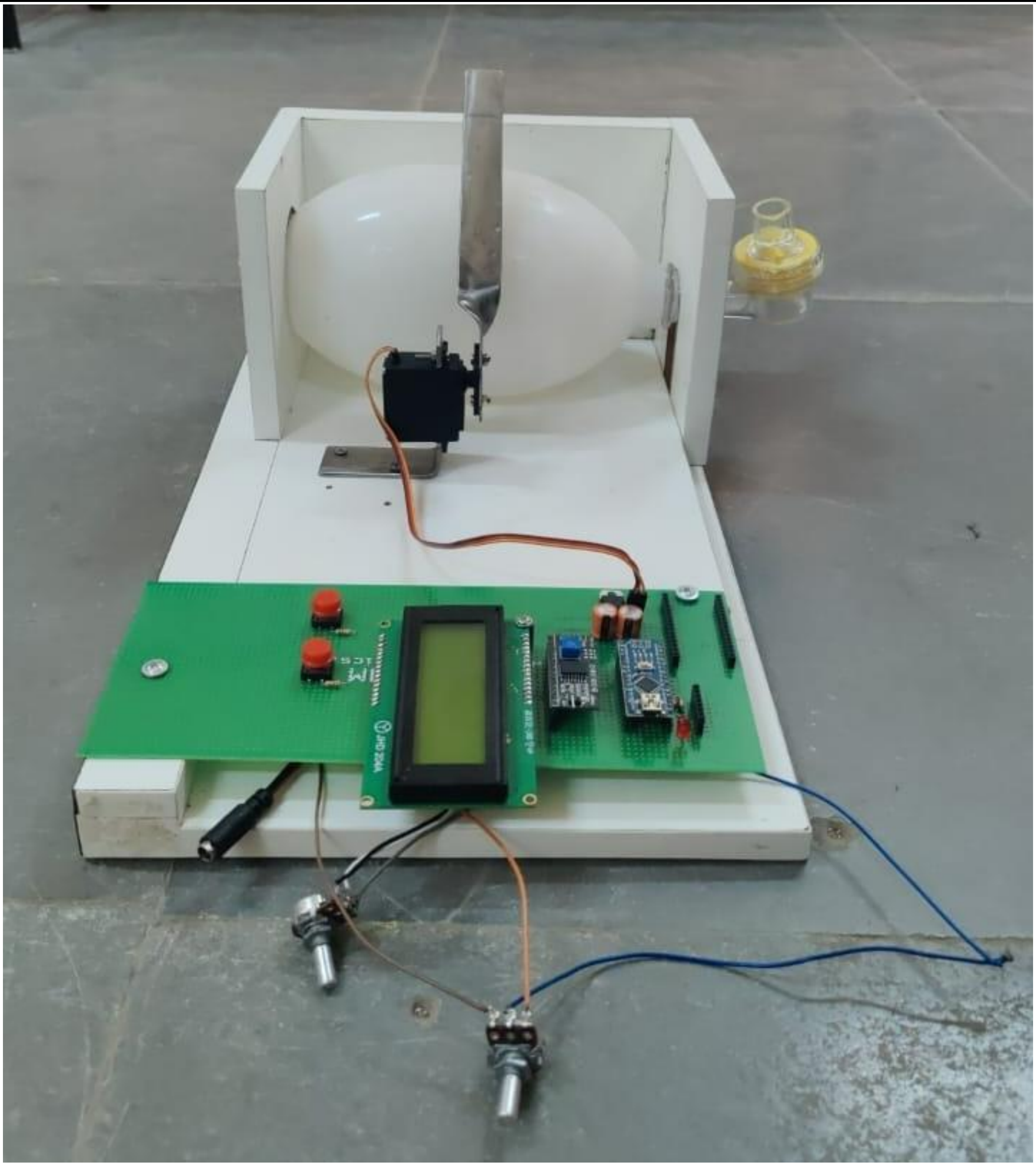
Fig 5.2: SERVO MOTER

CHAPTER 6

CONCLUSION

Since the start of the COVID-19 pandemic, researchers have been striving to help society face many problems caused by this pandemic. Among the recent initiatives, one has drawn the authors' attention: producing low-cost, open-source mechanical ventilators. The motivation comes from the worldwide shortage of mechanical ventilators in the treatment of COVID-19 patients—mechanical ventilators keep severely ill patients alive. This paper contributes to this initiative. This has detailed the construction of a functional, low-cost, and open-source mechanical ventilator. The authors' contribution to this topic aims to mitigate the effects of this worldwide ventilator's shortage—a shocking, unfortunate event that hits hard deprived areas. This has shown a numerical method that can monitor, in real-time, whether the patient has a healthy or unhealthy pulmonary condition. This useful yet straightforward numerical method opens up the possibility of applications in other mechanical ventilators as well. In summary, this paper contributes to both fronts—theory and practice. Alarms can be included in this project, using either an alarm screen or speakers, like the ones that alert clinicians when the pressure reaches some threshold values.

IMAGES



REFERENCES

1. Arduino Based Ventilator, Electronics Project, IoT project, Ai projects-based project, Home Automation project, Esp8266 based project, Esp32 project, Electronics, By Just Do Electronics, Updated Oct. 31, 2021 01:11 AM

2. Partially Rep Rapable automated open source bag valve mask-based ventilator, Author links open overlay panel Aliaksei Petsiuk ,Samantha Dertinger, Adam Pringle, Shane Oberloier, Joshua M.Pearce

3. Ranney, M.L.; Griffeth, V.; Jha, A.K. Critical Supply Shortages—The Need for Ventilators and Personal Protective Equipment during the Covid-19 Pandemic. *N. Engl. J. Med.* 2020, *382*, e41. [Google Scholar] [CrossRef] [PubMed]

4. Pons-Òdena, M.; Valls, A.; Grifols, J.; Farré, R.; Cambra Lasosa, F.J.; Rubin, B.K. COVID-19 and respiratory support devices. *Paediatr. Respir. Rev.* 2020, *35*, 61–63. [Google Scholar] [CrossRef] [PubMed]

5. Iyengar, K.; Bahl, S.; Raju, V.; Vaish, A. Challenges and solutions in meeting up the urgent requirement of ventilators for COVID-19 patients. *Diabetes Metab. Syndr. Clin. Res. Rev.* 2020, *14*, 499–501. [Google Scholar] [CrossRef] [PubMed]

6. Ferrante, L.; Fearnside, P.M. Protect Indigenous peoples from COVID-19. *Science* 2020, *368*, 251. [Google Scholar]