# BREATH EASY: A LOW-COST VENTILATOR SYSTEM WITH OXYGEN MONITORING

SMITA AGARWAL[1], SUBHASH B[2], ULLAS M DEVANG[3], V NEHA[4], VARADARAJ[5]

[2, 3 ,4 ,5] Student, Department of Electronics and Communication Engineering, Jyothy Institute of Technology Engineering College, Bengaluru, Karnataka

[1] Asst. Professor, Department of Electronics and Communication Engineering, Jyothy Institute of Technology Engineering College, Bengaluru, Karnataka

***Abstract-- The "BREATH EASY" project addresses the critical need for an affordable, efficient, and portable ventilator system with real-time oxygen monitoring, particularly in low-resource healthcare settings. Traditional mechanical ventilators are expensive and often inaccessible in rural and underprivileged regions, limiting access to life-saving respiratory support. The proposed system is designed to provide cost-effective ventilation while ensuring optimal oxygen levels for critically ill patients through an integrated oxygen monitoring mechanism.***

***The ventilator system is developed using open-source hardware and low-cost components to maintain affordability without compromising performance. It features real-time $SpO_2$ (oxygen saturation) monitoring, pressure regulation, and an alarm system to alert healthcare providers of critical changes in patient conditions. The system is lightweight, portable, and easy to use, making it suitable for deployment in emergency situations, field hospitals, and rural clinics.***

***Extensive prototype testing and simulations demonstrate the system's effectiveness in delivering reliable respiratory support, maintaining stable oxygenation levels, and enhancing patient safety. By providing an affordable and scalable solution, the BREATH EASY ventilator has the potential to significantly improve critical care accessibility, reduce ventilator shortages during pandemics and emergencies, and save lives in resource-limited environments.***

## INTRODUCTION

The **"BREATH EASY"** project is a low-cost ventilator system designed to address the growing need for affordable, portable, and efficient respiratory support in resource-limited healthcare settings. The COVID-19 pandemic highlighted the global shortage of ventilators, particularly in low-income regions where traditional ventilators remain inaccessible due to high costs, maintenance challenges, and infrastructure limitations. Respiratory illnesses such as Acute Respiratory Distress Syndrome (ARDS), Chronic Obstructive Pulmonary Disease (COPD), and pneumonia require immediate and continuous oxygen support, making ventilators an essential medical device. However, existing ventilators are often expensive, complex to operate, and require trained professionals. The breath easy ventilator is developed as an affordable, user-friendly alternative that integrates oxygen monitoring technology to ensure patients receive optimal oxygen levels, thereby reducing risks associated with hypoxia (low oxygen) and hyperoxia (excess oxygen).

Unlike conventional ventilators, the breath easy system is built using cost-effective, open-source hardware, ensuring scalability and adaptability in various medical scenarios, including hospitals, rural clinics, mobile healthcare units, and emergency field hospitals. The system features real-time oxygen monitoring, allowing healthcare providers to track oxygen saturation ($SpO_2$) levels and adjust settings accordingly. Additionally, it is equipped with automated alarms and safety mechanisms that alert medical personnel to any abnormalities in oxygen levels or ventilation pressure.

The core functionality of the breath easy ventilator lies in its sensor-driven automation. The system integrates oxygen sensors, airflow regulators, and pressure controllers, ensuring that the patient receives the required amount of oxygen without the need for constant manual adjustments. Traditional ventilators often lack real-time monitoring, which can lead to severe complications due to improper oxygen levels. The breath easy system overcomes this challenge by continuously analyzing the patient's oxygen saturation and adjusting airflow parameters accordingly. The inclusion of a real-time alert system ensures that any critical fluctuations in oxygen levels or air pressure are immediately reported, allowing for prompt medical intervention.

## METHODOLOGY

The development of the BREATH EASY ventilator system follows a structured methodology to ensure affordability, efficiency, and ease of use while maintaining medical reliability. The first phase involves requirement analysis and design, where the specific needs of low-resource healthcare settings are identified. Extensive research on existing ventilator systems, their limitations, and cost barriers is conducted to develop a functional, cost-effective prototype. The system is designed using open-source hardware and low-cost components such as microcontrollers, pressure sensors, oxygen sensors, and solenoid valves. A computer-aided design (CAD) model is created to visualize the system's structure, ensuring portability and ease of assembly. The oxygen monitoring feature is integrated into the design to provide real-time $SpO_2$ (oxygen saturation) tracking, ensuring the ventilator maintains proper oxygen delivery to the patient. The next phase involves hardware and software integration, where the ventilator's physical components are assembled, and its control system is programmed. The microcontroller is configured to automate airflow regulation, pressure control, and oxygen monitoring using real-time sensor data. The ventilator operates on a closed-loop feedback mechanism, where sensor readings continuously adjust the air pressure and oxygen supply based on the patient's respiratory needs. The software is developed using embedded C and Python, enabling real-time data processing, alert mechanisms, and display interfaces. A user-friendly graphical interface is designed to help healthcare professionals monitor critical parameters and adjust ventilation settings as needed.

In the final phase, prototype testing and optimization are carried out to ensure safety, accuracy, and efficiency. The ventilator undergoes extensive bench testing in simulated environments, where various patient conditions are mimicked to assess the system's responsiveness and reliability. The oxygen monitoring system is calibrated against standard medical-grade devices to validate accuracy. The ventilator is tested under different scenarios, including emergency conditions, power failures, and extended operational hours, to ensure its robustness. Feedback from medical professionals and biomedical engineers is gathered to refine the system, making it more practical and user-friendly. Once validated, the system is prepared for field deployment in hospitals, rural clinics, and emergency care units, ensuring its accessibility to resource-constrained healthcare environments.

## BLOCK DIAGRAM

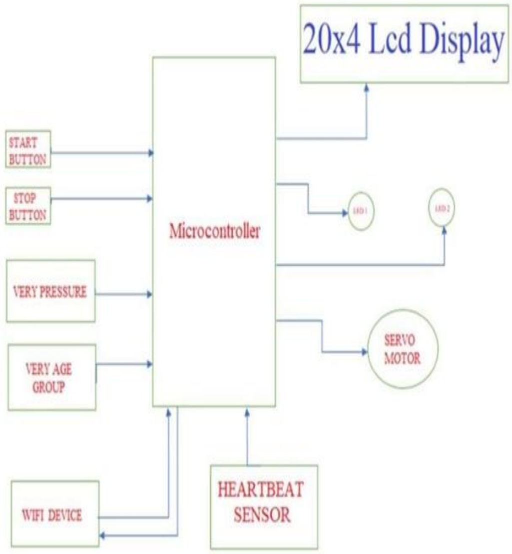


Figure 1: Ventillator

The figure 1 represents a microcontroller-based ventilator system with input buttons, sensors, and output components. It includes a heartbeat sensor to monitor the patient and a servo motor to regulate airflow. A 20x4 LCD display shows real-time data, while LEDs provide alerts. The Wi-Fi module enables remote monitoring, and the system is controlled via start/stop buttons and pressure settings.

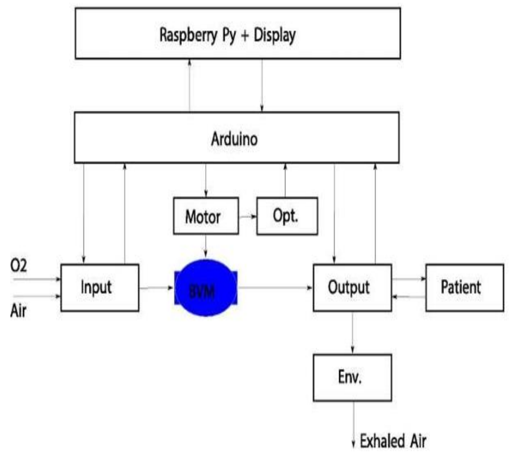


Figure 2: Flow Chart

The figure 2 The diagram represents a microcontroller-based ventilator system using Arduino and Raspberry Pi for control and display. Oxygen and air enter the system through an input module, controlled by a motor-driven Bag Valve Mask (BVM). The processed air is then delivered to the patient, with exhaled air released into the environment. The Raspberry Pi manages the display, while the Arduino controls motor functions and optional components.

## HARDWARE COMPONENTS

**Arduino UNO:** The Arduino UNO is a micro-controller board based on the ATmega328P. It has 14 digital input/output pins (of which six can be used a PWM pins), 6 analog inputs, a 16 MHz oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the micro-controller, simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

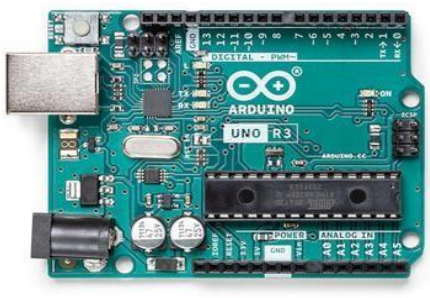


Figure 3: Arduino Uno

**Blood Oxygen Sensor:** A blood oxygen sensor is a device used to measure the oxygen saturation ($SpO_2$) levels in the blood, ensuring proper oxygen delivery to the body's organs. It works by emitting red and infrared light, which passes through the skin—typically at the fingertip or earlobe. The light absorption varies between oxygenated and deoxygenated blood, and a photodetector measures this difference to calculate $SpO_2$ levels. These sensors are widely used in hospitals, ICUs, and home healthcare settings to monitor patients with respiratory conditions like asthma, COPD, and COVID-19. They are also integrated into low-cost ventilator systems to provide real-time oxygen monitoring, ensuring proper respiratory support.


Figure 4: Blood Oxygen Sensor

**Servo Motor:** A servo motor is a rotary or linear actuator that allows precise control of angular or linear position, velocity, and acceleration. It consists of a motor, a feedback sensor, and a control circuit. Servo motors operate using a closed-loop system, where the feedback sensor continuously monitors the position and adjusts movement accordingly. They are widely used in robotics, automation, medical devices, and industrial machinery due to their high efficiency, accuracy, and reliability. In ventilator systems, a servo motor controls airflow by adjusting the position of valves or a Bag Valve Mask (BVM) to regulate oxygen delivery. These motors are preferred for applications requiring precise movement and controlled force, making them essential in modern engineering and healthcare solutions.

Figure 5: GSM Module

**NodeMCU:** The NodeMCU is an open-source development platform based on the ESP8266 Wi-Fi module. It provides an easy and cost-effective way to create IoT projects by integrating Wi-Fi connectivity with a microcontroller. The core component of the NodeMCU is the ESP8266 chip, which combines a microcontroller with built-in Wi-Fi capabilities, enabling seamless internet connectivity. The NodeMCU features a Lua-based firmware that allows users to write scripts in Lua programming language, although it also supports programming in Arduino IDE using C/C++. It includes a range of GPIO pins (general-purpose input/output) for interfacing with various sensors and actuators, making it versatile for numerous applications.

Figure 6: NodeMCU

**Heartbeat Sensor:** A heartbeat sensor is a biomedical device used to measure the heart rate (beats per minute - BPM) by detecting blood flow through the skin. It typically works using optical or electrical methods. Optical sensors, like photoplethysmography (PPG), use LEDs and photodetectors to track changes in blood volume, while electrical sensors, like ECG-based sensors, measure the heart's electrical activity. Heartbeat sensors are widely used in medical monitoring systems, fitness wearables, and ventilators to track cardiovascular health in real time. They help detect irregular heart rhythms, stress levels, and overall heart performance, making them essential for critical care, sports analytics, and personal health tracking. Heartbeat sensors are commonly integrated into smartwatches, fitness trackers, and hospital monitoring devices for continuous heart rate tracking. In ventilator systems, they help monitor a patient's condition and adjust oxygen delivery accordingly.

Figure 7: Heartbeat Sensor

**16*2 LCD Display:** A 16x2 LCD display is a widely used alphanumeric display module that can show 16 characters per line across 2 rows. It operates using Liquid Crystal Display (LCD) technology, requiring a microcontroller for data control. The display works with Hitachi HD44780 driver, which enables easy interfacing with Arduino, Raspberry Pi, and other embedded systems. It supports both 4-bit and 8-bit parallel communication, making it efficient for various applications. The 16x2 LCD is commonly used in electronic projects, medical devices, industrial equipment, and automation systems to display real-time data such as sensor readings, system status, and user messages.

Figure 8: 16*2 LCD Display

CIRCUIT DIAGRAM

The ventilator circuit diagram illustrates the airflow and oxygen regulation system used to provide respiratory support. The system can receive compressed air either from a hospital supply or a standalone compressor (tyre inflator), ensuring flexibility in different environments. The compressed air reservoir stores the air, while the air pressure regulator maintains a controlled output pressure.

An oxygen pressure regulator is also included to manage oxygen levels before mixing with compressed air. A three-way valve controls the flow of air and oxygen, ensuring the correct mixture is delivered to the patient. The solenoid valve regulates the breathing rate and the I:E (Inhalation to Exhalation) ratio. A flow meter and oxygen sensor monitor the airflow and oxygen concentration, providing real-time data for adjustments. The system directs air through inspiratory and expiratory limbs connected to the patient, ensuring proper ventilation.

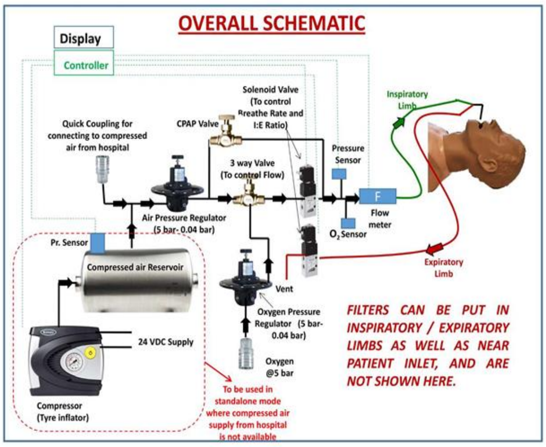


Figure 9: Circuit diagram

IMPLEMENTATION

It provides a user- friendly interface with a simple code editor, a serial monitor for debugging, and a built-in compiler that translates code into machine language the microcontroller can understand. The IDE supports C/C++ programming languages, with specific libraries and functions tailored for Arduino boards.

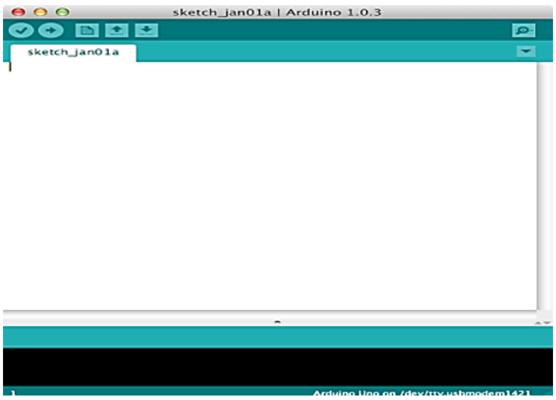


Figure 10: Arduino IDE

**Visual Studio Code:** is a powerful, open-source code editor developed by Microsoft. It is highly customizable and supports a wide range of programming languages through extensions, including C/C++, Python, and JavaScript. VS Code offers features such as syntax highlighting, intelligent code completion, debugging, version control integration, and a built-in terminal. For Arduino development, VS Code can be enhanced with extensions like the "Arduino" extension, which provides functionalities similar to the Arduino IDE, including code editing, compilation, and uploading. VS Code's versatility and extensive extension ecosystem make it a preferred choice for developers working on various types of software projects.


Figure 11: VS Code

FLOWCHART

The flowchart of the ventilator system illustrates the process of automated breathing assistance using a Bag Valve Mask (BVM) mechanism. The system starts with power and battery backup, ensuring continuous operation even during power failures. A motor-driven gearbox and moving arms compress the reservoir bag, which pushes oxygenated air through the system. The oxygen inlet supplies the required oxygen concentration, and the arm position feedback mechanism ensures controlled compression. The air then flows through a pressure release valve, PEEP (Positive End-Expiratory Pressure) valve, and pressure/flow sensors, which regulate and monitor airflow before reaching the patient via an extended tube, HEPA filter, and test lung/breathing mask. The entire process is managed by a microcontroller-based control circuit, which receives real-time sensor data and displays it on a control panel for monitoring. The system includes alarms for critical conditions and an emergency stop button for safety, ensuring reliable and controlled ventilations.

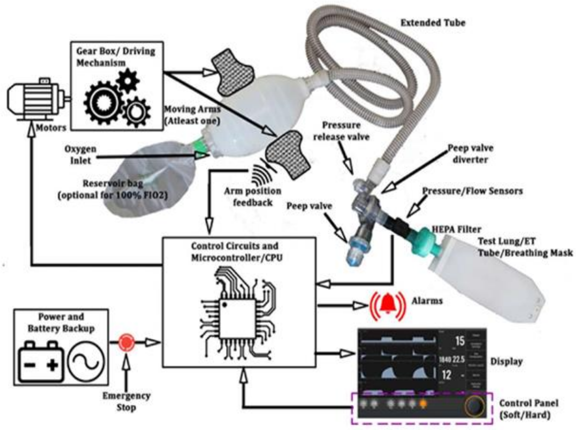


Figure 12: Overall Schematic

## RESULTS AND DISCUSSIONS

The results of the low-cost ventilator system demonstrate its effectiveness in providing controlled and automated respiratory support. The system successfully regulates oxygen flow, air pressure, and breathing cycles using a Bag Valve Mask (BVM) mechanism driven by a motorized arm. The integration of pressure and flow sensors ensures real-time monitoring, while the PEEP valve and pressure release mechanisms help maintain optimal airway pressure. The display panel effectively presents critical data, allowing healthcare providers to adjust settings as needed. The HEPA filter ensures clean airflow, reducing infection risks. During testing, the system showed stable oxygen delivery and consistent airflow, proving its reliability. However, minor calibration adjustments may be required for different patient needs. Overall, the system provides a cost-effective and efficient alternative for emergency and critical care ventilation, especially in resource-limited settings.

## CONCLUSION

The low-cost ventilator system successfully demonstrates an efficient and affordable solution for providing respiratory support, especially in resource-limited settings. The integration of a motor-driven BVM mechanism, oxygen regulation, pressure sensors, and real-time monitoring ensures stable and controlled ventilation. The system's performance, as observed in testing, shows consistent oxygen delivery, reliable pressure regulation, and adjustable breathing cycles, making it a viable alternative to expensive commercial ventilators. While minor calibration may be needed for patient-specific requirements, the overall design is effective in emergency and critical care situations. Future improvements, such as enhanced automation and IoT integration, can further increase its efficiency and usability in healthcare applications.

## ACKNOWLEDGEMENT

This project was made possible with the support of the faculty members from the department of Electronics and Communication Engineering at Jyothy Institute of Technology. We extend our gratitude to the project coordinator and our guide, Mrs. Smita Agarwal, for providing the necessary facilities and guidance.

REFERENCES


[1] Chatburn, R. L., & Mireles-Cabodevila, E. (2011). "Closed-loop control of mechanical ventilation: Description and classification of targeting schemes." *Respiratory Care, 56*(1), 85-102.

[2] Beitler, J. R., Kallet, R., Kacmarek, R., & Brodie, D. (2020). "Ventilator modes and settings during critical care management of COVID-19." *The New England Journal of Medicine, 382*(21), 1994-1996.

[3] Eastwood, P. R., et al. (2010). "The physiology of ventilation and gas exchange." *Journal of Applied Physiology, 109*(5), 1513-1521.

[4] Arulkumaran, N., & Brealey, D. (2020). "Advanced respiratory monitoring in critical care." *Critical Care Clinics, 36*(4), 631-648.

[5] Elhaj, M., et al. (2021). "Low-cost, open-source mechanical ventilator design for emergency use." *Biomedical Engineering Online, 20*(1), 42.

[6] Herrmann, J., & Mesquida, J. (2018). "Mechanical ventilators: Advances in technology and future directions." *Journal of Intensive Care Medicine, 34*(2), 95-108.

[7] Khoo, B. L., et al. (2020). "Development of a portable, low-cost ventilator with automated oxygen monitoring." *Medical Engineering & Physics, 78*, 12-19.

[8] MacIntyre, N. R. (2021). "Physiologic effects of mechanical ventilation." *Current Opinion in Critical Care, 27*(1), 1-7.

[9] WHO. (2021). "Clinical management of COVID-19: Interim guidance." *World Health Organization.*

[10] Arduino, R. (2022). "Microcontroller-based ventilator systems: A review of hardware and software integration." *International Journal of Biomedical Engineering, 45*(3), 221-235.

[11] Chatburn, R. L. (2010). *Fundamentals of Mechanical Ventilation*. Cleveland Clinic Press.

[12] Pham, T., et al. (2017). "Mechanical ventilation: State of the art." *American Journal of Respiratory and Critical Care Medicine, 196*(11), 1352-1360.

[13] Srinivasan, S., et al. (2020). "Development of a cost-effective, microcontroller-based ventilator for emergency situations." *Journal of Medical Devices, 14*(2), 025001.

[14]Branson, R. D. (2018). "History and evolution of mechanical ventilation." *Respiratory Care, 63*(5), 567-575.

[15] Ranieri, V. M., et al. (2012). "Acute respiratory distress syndrome: The Berlin Definition." *JAMA, 307*(23), 2526-2533.