Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.776839

# PRECISION CONTROLLED VENTILATION USING AMBU BAG FOR MACHINE LEARNING DATASET CREATION

**Aditya Kumar**
School of Electrical and Electronics Engineering (SELECT) Vellore Institute of Technology Vellore, India

**Vishnu P**
School of Electrical and Electronics Engineering (SELECT) Vellore Institute of Technology Vellore, India

**Dr. Washima Tasnin**
School of Electrical and Electronics Engineering (SELECT) Vellore Institute of Technology Vellore, India

**Abstract—**Providing efficient ventilation to patients in emer- gency situations can be challenging, particularly in settings with limited resources. This research aims to address this issue by proposing an automated AMBU bag system that can determine the ventilation rate based on the patient's pulse rate and blood oxygen level. The proposed system consists of a microcontroller, a sensor, cloud services, a stepper motor driver, and a stepper motor. The system reads the patient's pulse rate and blood oxygen level using a sensor, processes the data using an equation, and adjusts the stepper motor's ventilation rate accordingly. Another aim of the project is to create a system that can help maintain a dataset using the integration of AWS for future research. By creating a dataset for further research, this project can lead to the development of more advanced machine learning models that can improve the efficiency of ventilation systems. Furthermore, the proposed system can be customized to fit the specific needs of different healthcare settings, allowing for widespread adoption of this technology. Overall, the automated AMBU bag system has the potential to greatly improve patient care in emergency situations.

**Index Terms—**AMBU Bag, Machine Learning, MAX30100 Sensor, 3D Printing, Ventilator, Dataset, Raspberry Pi, Amazon Web Services

## I. INTRODUCTION

Mechanical ventilation is a critical component of emergency medical care and plays a vital role in saving the lives of pa- tients with respiratory distress. It is used to support or control breathing in individuals with various respiratory problems such as Acute Respiratory Distress Syndrome (ARDS), Chronic Obstructive Pulmonary Disease (COPD), Pneumonia, Bron- chitis, Asthma, Respiratory Muscle Weakness or Paralysis, Neuromuscular conditions such as ALS (Amyotrophic Lateral Sclerosis), and Sleep Apnea. The aim of this project is to provide a novel solution for emergency mechanical ventila- tion that addresses the limitations of traditional mechanical ventilators.

The aim of the project is to develop a system that can collect data to aid in the further research of machine learning for automated AMBU bags. Automated AMBU bags can play a crucial role in emergency situations where patients require mechanical ventilation. However, current systems lack the precision and accuracy required for optimal patient care. The proposed system uses a stepper motor to control the ventilation rate of the AMBU bag based on the patient's pulse rate and blood oxygen level. The data collected by the system can be used to train machine learning models that can improve the accuracy and reliability of automated AMBU bag systems.

The proposed system comprises of several components, including a Raspberry Pi 4 Model B microcontroller, a MAX30100 sensor, AWS(Amazon Web Services) Cloud Ser- vices, a stepper motor driver (TB6600), and a stepper motor. The MAX30100 sensor reads the oxygen level and pulse rate of the patient and sends the data to AWS IoT Core. An AWS Lambda function is then created to process the ventilation rate using a PID equation. The ventilation rate, along with the corresponding pulse rate and blood oxygen value, is then published on an AWS DynamoDB table. The Raspberry Pi reads the ventilation rate from the DynamoDB table, and the rate of the stepper motor is updated based on the value fetched from the table. The stepper motor starts at the initial rate of 12, and at no point should it stop rotating. The value from the DynamoDB is retrieved every 30 seconds to ensure the system is up to date with the latest data.

The proposed system is a proof-of-concept that demon- strates the feasibility of using a stepper motor to control the ventilation rate of an AMBU bag based on the patient's pulse rate and blood oxygen level. The system is intended to collect data that can be used for further research in the area of machine learning for automated AMBU bags. By collecting data in a centralized location using AWS, the system can provide a large and diverse dataset that can be used to develop and validate new machine learning algorithms, evaluate the performance of existing algorithms, and identify areas for improvement in automated AMBU bag systems.

The lack of available data has been a significant obstacle in the development of machine learning algorithms for automated AMBU bags. The proposed system aims to overcome this obstacle by providing a large and diverse dataset that can be

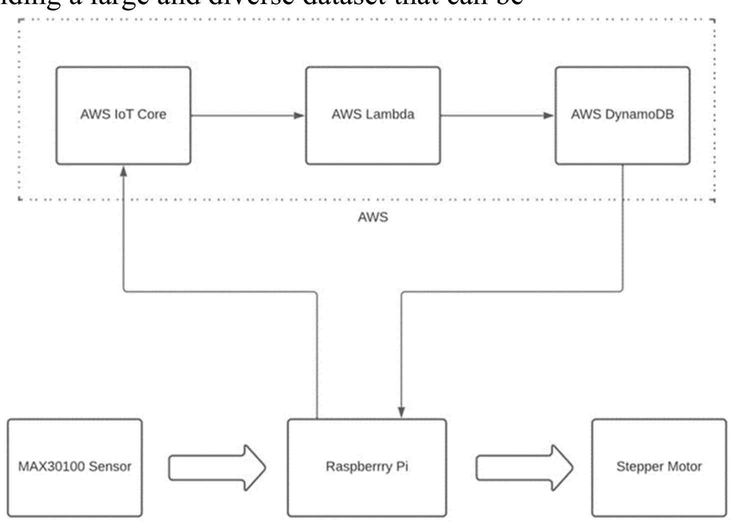


Fig. 1. Block Diagram of the Project

used for research purposes. This dataset can help researchers to develop and validate machine learning algorithms that can improve the accuracy and reliability of automated AMBU bag systems. With the ability to collect data in real-time, the proposed system can provide valuable insights into how patient conditions change over time and how AMBU bag systems can adapt to these changes.

The proposed system has several potential benefits, in- cluding improved patient outcomes, reduced healthcare costs, and increased efficiency in emergency care. By providing automated AMBU bag systems that can adapt to patient condi- tions in real-time, healthcare professionals can focus on other critical tasks, reducing the burden on healthcare workers and improving patient outcomes. Moreover, the proposed system can help reduce healthcare costs by reducing the need for manual ventilation and by minimizing the risk of complica- tions associated with mechanical ventilation. Additionally, the proposed system can increase the efficiency of emergency care by providing automated AMBU bag systems that can respond to patient conditions quickly and accurately.

In conclusion, the proposed system demonstrates the fea- sibility of using a stepper motor to control the ventilation rate of an AMBU bag based on the patient's pulse rate and blood oxygen level. The system aims to collect data that can be used to further research in the area of machine learning for automated AMBU bags. With the ability to collect data in real-time, the proposed system can provide valuable insights into how patient conditions change over time and how AMBU bag systems can adapt to these changes. The potential benefits of the proposed system include improved patient outcomes, reduced healthcare costs, and increased efficiency in emergency

The overall architecture of the system is shown through a block diagram in Figure 1. The MAX30100 sensor is connected to the Raspberry Pi through the I2C interface, and sensor data is read using the appropriate Python libraries. The Raspberry Pi is also connected to the internet through Wi-Fi, allowing it to transmit and receive data from AWS IoT Core and DynamoDB. The TB6600 stepper motor driver and stepper motor are connected to the Raspberry Pi through the GPIO pins, and stepper motor control is implemented in software using Python. AWS Lambda hosts Lambda function that analyses and process the inputs gathered from the MAX30100 sensor, i.e., the blood oxygen level and pulse rate of the patient in real time. The database hosts the patient's data collected and the speed of the stepper motor that is decided by the Lambda function.

The software used to control the system was developed in Python, taking advantage of the wide range of libraries and tools available for the Raspberry Pi. The software was designed to read sensor data, transmit it to AWS IoT Core, retrieve ventilation rate data from DynamoDB, and adjust the speed of the stepper motor based on the retrieved value. The software also incorporates error handling and logging functionality to ensure the reliability and robustness of the system.

## II. RELATED WORK

Mechanical ventilators are essential devices used to support patients with respiratory failure, especially during pandemics such as COVID-19. There are several types of mechanical ventilators available, including bag valve mask (BVM) venti- lators, electronically controlled mobile ventilators, and internet of things (IoT)-based ventilators. In this section, we provide an overview of ten research papers that present novel approaches to designing and implementing mechanical ventilators.

Van Tu Duong et al. [1], presents a modeling process for a bag valve mask (BVM) ventilator system by analyzing the relationship between the gripper angle and the exhausted air volume of an AMBU bag. The authors conducted experiments to collect data on the two variables and found a non-linear re- lationship between them. A regression analysis was performed to fit a linear equation to the data and the results showed a high coefficient of determination and low error rates. The authors also discussed the potential applications of their research in improving the control performance of BVM ventilators and reducing errors in future studies. Overall, the paper provides a valuable contribution to the field of respiratory therapy and mechanical ventilation.

Rohan Lal Kshetry et al. [2], discuss the design and simulation of a low-cost, portable mechanical ventilator as an alternative to the manually operated AMBU bag during pandemics causing acute respiratory distress syndrome. The proposed ventilator uses a robust kinematic linkage and closed- loop monitoring strategy and can be fabricated using conven- tional CNC machine tools or 3D printing. The article also dis- cusses the incorporation of AI, IoT, and predictive monitoring for diagnostics, making it a hygienic, easily assembled, and disassembled alternative. The authors highlight the importance of the proposed ventilator in times of pandemics, especially in developing and underdeveloped regions, where access to ventilators is limited.

Rameshwar Bhaginath Kale et al. [3], Describes a low- cost, homebrew ventilation system that may be built with components that are inexpensively available and an Arduino microcontroller. The authors conduct a literature review on the difficulties low-income nations experience in obtaining appropriate medical supplies as well as the significance of mechanical ventilation in the treatment of respiratory failure in COVID-19 patients. They go over the ventilator's design and construction, including how a blood oxygen sensor is used to track the patient's oxygen saturation levels. The authors present experimental findings that show the ventilator's effectiveness in preserving constant oxygen saturation levels in a test participant. The study emphasises the promise of inexpensive, open-source ventilator devices in helping patients in low-income nations with life-saving therapy during the COVID-19 pandemic.

Mohsen Al Husseini et al. [4], present the development and design of a low-cost, portable ventilator for use in emergency and low-resource settings. The paper describes the design process, starting with initial concept development and evaluation of different mechanisms for compressing the bag valve mask (BVM) used in ventilation. The authors ultimately chose a cam-based mechanism, which was found to be more space-efficient and lower in power requirement than a roller- chain mechanism. Two prototype designs are presented, along with experimental results demonstrating the device's ability to deliver accurate and repeatable tidal volumes. The paper also discusses the implementation of control algorithms to ensure safe and effective operation of the device. Overall, the research presents a promising solution to the need for low-cost, portable ventilators in emergency and resource-limited setting.

Onintza Garmendia et al. [5], present a unique design for a non-invasive, low-cost ventilator is presented. It may be made from readily available components and an Arduino microcontroller. The authors conduct a literature review on the issues surrounding access to proper medical equipment that under-resourced areas face, particularly in light of the COVID-19 pandemic. They go over the ventilator's design and construction, which includes controlling pressure and venti- lation cycles with an Arduino microcontroller and pressure transducers. The authors present experimental findings that show how well the ventilator performs in keeping stable ventilation in both healthy volunteers and a simulated patient. The study emphasises how open-source, low-cost ventilator systems could help address the lack of medical supplies in underdeveloped areas, especially in times of a public health emergency like the COVID-19 pandemic.

Mukaram Shahid et al. [6], discusses a prototype system designed to monitor the breathing pattern of COPD (chronic obstructive pulmonary disease) patients. The system uses a thermistor and an MQ-135 sensor to detect temperature and CO2 level changes in the breath, respectively. The signals from the sensors are amplified and filtered through a bandpass filter. The pressure inside the mask is controlled by a stepper motor- based valve, and the resulting pressure waveform is observed linearly according to the motor's position. The system was tested on a COPD patient in a local medical clinic, with a registered medical doctor supervising the test. The output signals from the sensors were shown using a CRO, and it was observed that the MQ-135 sensor did not show any output when the patient inhaled. The overall system was found to be reliable.

Ahmed Ibarra Abboudi et al. [7], describe the design and implementation of the ventilator, which uses an Arduino microcontroller and a set of sensors to measure various phys- iological parameters, including breathing rate, tidal volume, and oxygen saturation. The ventilator also features an alarm system to alert healthcare providers of any abnormalities in these parameters. The authors tested the ventilator on a patient simulator and demonstrated that it could effectively deliver controlled ventilation. They also evaluated the device's safety features, including an emergency stop button and a pressure relief valve to prevent barotrauma. The authors' approach offers a potential solution to the shortage of ventilators during the COVID-19 pandemic. The device's low cost and porta- bility make it an attractive option for emergency use in low- resource settings or during disasters. However, further testing and validation of the device are necessary before widespread adoption.

Leonardo Acho et al. [8], discuss the design and con- struction of a low-cost, open-source mechanical ventilator aimed at mitigating the effects of the worldwide shortage of mechanical ventilators during the COVID-19 pandemic. The authors describe the construction of the device and present experimental data showing the efficient control of three main variables of the ventilator, namely respiratory frequency, the ratio of inspiration-to-expiration, and the air volume supplied to the patient. The authors also present a numerical method that can monitor, in real-time, whether the patient has a healthy or unhealthy pulmonary condition. The authors acknowledge that the study has some intrinsic limitations, and it is unclear whether the ventilator complies with regulatory agency regu- lations such as MHRA and FDA. The authors conclude that their contribution to this initiative aims to mitigate the effects of the ventilator shortage, which hits deprived areas hard, and open up the possibility of applications in other mechanical ventilators.

Mahdi Robati et al. [9], present the design and testing of an affordable mechanical ventilator that automates the manual squeezing of a bag valve mask for emergency respiratory support. The device uses a stepper motor, belt drives, and spur gears to drive two pushing handles and a flow sensor for closed-loop control of air volume based on lung parameters. The system also includes pressure sensors for safety and an estimation of test lung parameters using flow and pressure data. The estimated compliance of the test lung matched its reported compliance, but the resistance was higher due to the long tube used to connect the prototype to the test lung.

Paolo Pelosi et al. [10], discuss the potential benefits of personalized mechanical ventilation for patients with acute res- piratory distress syndrome (ARDS). The approach is based on lung physiology, morphology, ARDS etiology, lung imaging, and identification of biological phenotypes. Personalized me- chanical ventilation may improve and individualize future me- chanical ventilation practice, but additional research is needed before implementing it at the bedside of ARDS patients. The article emphasizes the importance of considering patient-centered outcomes and the uncertainty surrounding the effects of physiological manipulations. The future research agenda includes characterizing biomarker profiles and responses to specific treatments, using mechanical power and PL, and identifying patient phenotypes.

Overall, the ten papers provide valuable insights into the de- sign, development, and implementation of mechanical ventila- tors in healthcare settings. They suggest that novel approaches, such as modeling processes, robust kinematic linkages, IoT- based systems, and real-time sensors, could improve the ef- ficiency, cost, mean time between failure, and affordability of mechanical ventilators. Further research is necessary to validate and test these approaches and to explore their potential implications for improving the quality of life and resource utilization in healthcare settings.

## III. METHODOLOGY

The proposed design for the emergency mechanical ven- tilator system consists of a combination of hardware and software components that work together to provide efficient and effective respiratory support for patients. It is crucial to break down the project into different parts to gain a full understanding of how the system operates. The importance of this project is the ease at which each component of the project can be separated from the main project and improved. Fig. 1 shows the components of the project.

### A. System Design

The system design of the smart oxygen monitoring system involves the integration of several hardware and software components to achieve the desired functionality. The use of 3D printing technology to create the side push mechanism and the selection of the stepper motor and TB6600 stepper motor driver were key components of the design. The Raspberry Pi microcontroller serves as the central control unit for the system, providing the necessary processing power and con- nectivity to read and transmit sensor data, retrieve ventilation rate data from AWS DynamoDB, and control the stepper motor. Figure 2 shows the Raspberry Pi used in the project. To facilitate the proper placement of the AMBU bag, 3D printed clamps were designed to hold the bag securely in place. The clamps were printed in two parts and then assembled, as shown. Figure 3 shows the AMBU Bag used in the project. One side of the AMU Bag is used to connect the nose-mask to the patient and other is used for oxygen/air supply. The intake valve can either be connected to an oxygen cylinder or left open for normal air to rush in when the bag is commpressed, due to pressure difference.

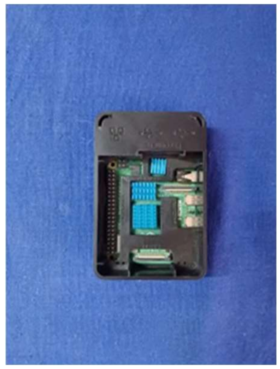

Fig. 2. Raspberry Pi Model 4B

To implement the two side push mechanism, 3D printing technology was used to create the required parts. The 3D printed objects were designed to ensure smooth and accurate movement of the AMBU bag while minimizing friction and wear. The 2 side push mechanism consists of two arms, each attached to the stepper motor, through a rod, and contacts the AMBU bag. The pusher arms are symmetrically positioned on either side of the bag, ensuring balanced and consistent

**Fig. 3. AMBU Bag**

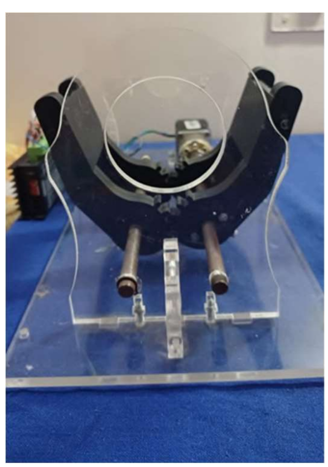

**Fig. 4. 3D Clamps from the Side**

movement. Figure 4 shows the 3D clamps from the side and Figure 5 shows the same hardware prototype from a top view. The stepper motor was selected for its precise control over speed and position, as well as its ability to rotate continuously without requiring manual intervention. The TB6600 stepper motor driver was chosen for its compatibility with the Rasp- berry Pi and its ability to provide the necessary power to drive the stepper motor. To complete the system design, the software was developed in Python, taking advantage of the wide range of libraries and tools available for the Raspberry Pi. The software was designed to read sensor data, transmit it to AWS IoT Core, retrieve ventilation rate data from DynamoDB, and adjust the speed of the stepper motor based on the retrieved value. The software also incorporates error handling and logging functionality to ensure the reliability and robustness
of the system.

In summary, the system design of the smart oxygen mon-

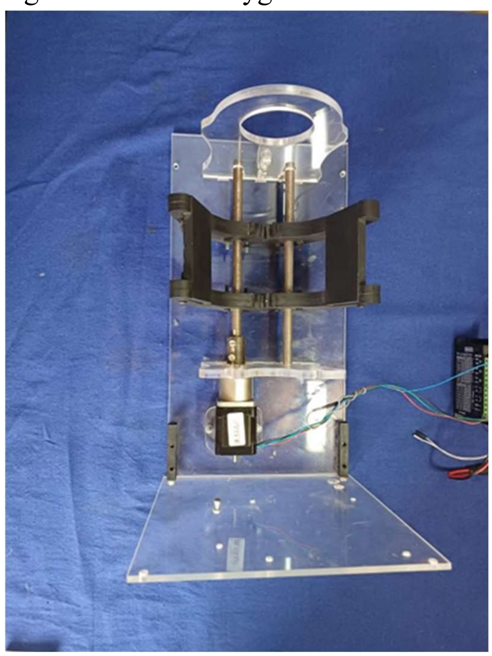

Fig. 5. 3D Clamps from the Top

itoring system involves the integration of several hardware and software components to achieve the desired functionality. The architecture was designed to ensure reliable and accurate sensor data collection, transmission to AWS IoT Core, retrieval of ventilation rate data from DynamoDB, and precise control over the stepper motor. The use of 3D printing technology to create the side push mechanism, the selection of the stepper motor and TB6600 stepper motor driver, and the use of Python software were critical components of the design

B. Sensor Data Collection and Analysis

Sensor data collection and analysis is a critical component of the project as it enables the system to monitor the patient's vital signs and adjust the stepper motor speed to maintain the desired oxygen saturation level. For this purpose, we are using the MAX30100 sensor, which is a pulse oximetry and heart- rate monitor sensor.

The MAX30100 sensor has two LEDs and a photodetector that detect the amount of light transmitted through the patient's finger. The amount of light absorbed by the patient's blood is then used to calculate the oxygen saturation level and pulse rate. The sensor outputs an analog signal that is converted to a digital signal by the Raspberry Pi's ADC. Figure 6 shows the MAX30100 sensor and its connections, as used in the project. The data from the MAX30100 sensor is filtered using signal processing techniques such as low-pass and moving average filters. The filtered data is then analyzed to calculate the pulse rate and oxygen saturation level. To calculate the appropriate speed for the stepper motor, we use a proportional- integral-derivative (PID) controller. The PID controller is a feedback control system that adjusts the output based on the error between the desired setpoint and the measured process

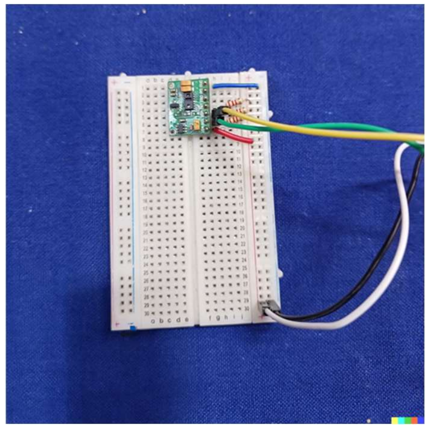

Fig. 6. MAX30100 Sensor

variable. In this case, the setpoint is the desired oxygen saturation level, and the process variable is the actual oxygen saturation level measured by the sensor. The equation for the PID controller is set using:

ability to maintain the patient's oxygen saturation level within the desired range. The implementation of the PID controller ensures that the system responds appropriately to changes in the patient's condition by adjusting the stepper motor speed.

C. AWS Cloud Services

AWS cloud services were a critical component of this project, enabling the integration of the Raspberry Pi with various cloud-based services, including IoT Core, Lambda, and DynamoDB. The use of AWS services allowed for real- time data processing, storage, and analysis, resulting in a more efficient and scalable system.

The sensor data collected from the MAX30100 sensor is sent to the Raspberry Pi for further processing. Once the data has been filtered and analyzed to determine the appropriate speed for the stepper motor based on the PID algorithm, the data is sent to AWS IoT Core. An AWS Lambda function is then created to take the value in json format from IoT Core and calculate the appropriate speed for the stepper motor based on the PID algorithm.

The Lambda function receives the sensor data in real-time and calculates the output based on the current error and the PID parameters. The output is then sent and stored in AWS DynamoDB, corresponding to the patient's blood oxygen level and pulse rate, from where the Raspberry Pi retrieves the stepper motor speed. The integration of AWS services with the Raspberry Pi has allowed for a more streamlined and efficient system, enabling real-time data processing and analysis.

One of the main advantages of using AWS services in this
project is the ability to scale the system to meet changing demands. As the number of patients monitored increases,
output = Kp ∗ error + Ki ∗
where,

error(t )dt  + Kd ∗ dt (error)
0        (1)
the system can be easily scaled up to handle the additional load. This scalability is made possible by the use of cloud- based services, which provide virtually limitless storage and

- Kp is the proportional gain,
- Ki is the integral gain,
- Kd is the derivative gain,
- error is the difference between the desired setpoint and the measured process variable,
- Ki is the integral gain,
- t is the current time.

The proportional gain adjusts the output based on the current error, the integral gain adjusts the output based on the accumulated error over time, and the derivative gain adjusts the output based on the rate of change of the error.

To implement the PID controller, the sensor data is sent to AWS IoT Core, where an AWS Lambda function calculates the appropriate speed for the stepper motor based on the PID algorithm. The Lambda function receives the sensor data in real-time and calculates the output based on the current error and the PID parameters. The output is then sent and stored in AWS DynamoDB, corresponding to the patient's blood oxygen level and pulse rate, from where the Raspberry Pi retrieves the stepper motor speed.

In conclusion, the MAX30100 sensor, along with the signal processing and PID controller, plays a vital role in the system's computing power.

Another advantage of using AWS services is the ease of integration with other cloud-based services. The use of AWS IoT Core, Lambda, and DynamoDB in this project allowed for a seamless integration of various cloud-based services, resulting in a more robust and efficient system.

In summary, the use of AWS cloud services in this project has enabled real-time data processing, storage, and analysis, resulting in a more efficient and scalable system. The integration of various cloud-based services, including IoT Core, Lambda, and DynamoDB, has allowed for a seamless and robust system that can easily scale to meet changing demands. The use of AWS services has been essential in achieving the goals of this project, enabling real-time monitoring of vital signs and adjusting the stepper motor speed to maintain the desired oxygen saturation level.

**D. Stepper Motor Control**

The Stepper Motor plays a crucial role in the project as it drives the ventilation system. The purpose of the ventilation system is to rotate the 3D printed pieces to increase the airflow from the AMBU bag to the patients' lungs in an emergency situation. The stepper motor controls the speed and frequency at which the air is pushed in and out of the lungs of the patient. Figure 7 shows the stepper motor used in the project. It is attached to a single rod that rotates both clockwise and anit-clockwise depending on the signal given it.

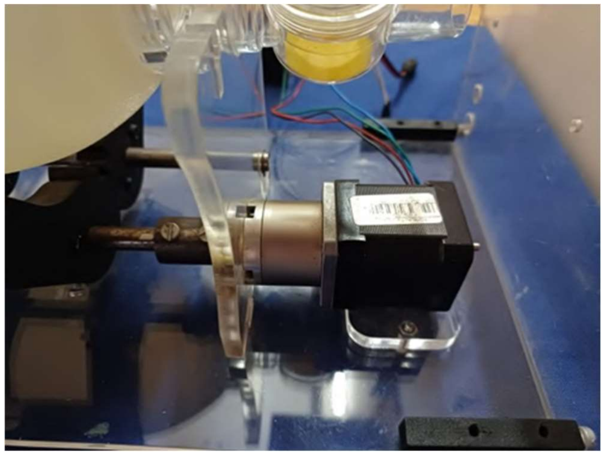

Fig. 7. Stepper Motor

In this project, the stepper motor is controlled using a Raspberry Pi, and AWS IoT is used to store and analyze data. The stepper motor used in this project is a unipolar stepper motor, which has four coils and is controlled using a sequence of high and low signals. The direction of the motor is controlled by the direction signal, and the speed of the motor is controlled by the frequency of the pulse signal. The Raspberry Pi is used to generate the signal required to control the stepper motor, and the frequency and direction of the signal can be adjusted based on the desired speed and direction of the motor. The stepper motor is connected to the Raspberry Pi using two pins, the DIR pin, and the STEP pin. The DIR pin is used to control the direction of the motor, and the STEP pin is used to control the frequency of the pulse signal. When the STEP pin is pulsed, the motor moves one step in the specified direction. The frequency of the pulse signal can be adjusted to control the speed of the motor. The higher the frequency of the pulse signal, the faster the motor will move. The stepper motor is controlled using a program written in Python that runs on the Raspberry Pi. The program generates a pulse signal with a specified frequency and direction, and the motor moves accordingly. The program is designed to read the desired ventilation rate from a database hosted on AWS DynamoDB and adjust the motor's speed accordingly. The stepper motor is used to drive the ventilation system and is crucial for the performance of the entire project. Figure 8 shows us the stepper motor driver that we have used in the project.

To control the stepper motor, a stepper motor driver is used to amplify and interpret the signals generated by the Raspberry Pi. The stepper motor driver acts as an interface between the Raspberry Pi and the stepper motor, allowing the Raspberry

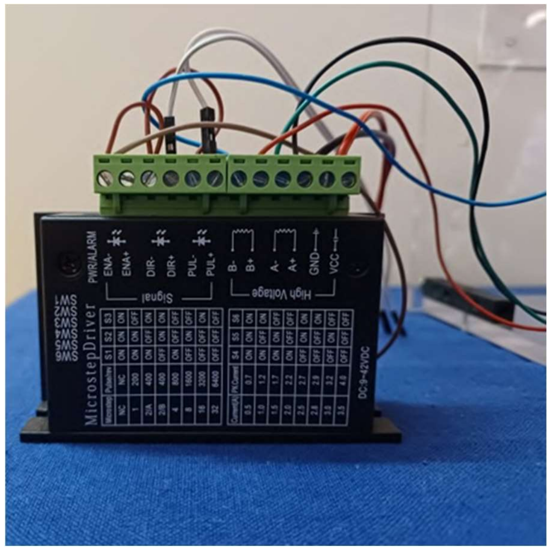


Fig. 8. Stepper Motor

Pi to control the motor more precisely. The stepper motor driver is necessary because the Raspberry Pi is not capable of supplying enough current to drive the stepper motor directly. The stepper motor driver is designed to receive signals from the Raspberry Pi and convert them into a form that can drive the motor. The driver is also responsible for controlling the amount of current that flows through the coils of the motor, which is necessary to control the speed and torque of the motor. Without the stepper motor driver, the Raspberry Pi would not be able to control the stepper motor accurately, and the motor's performance would be unpredictable. Therefore, the stepper motor driver and the stepper motor need to work together to ensure the proper functioning of the ventilation system.

### E. Final Setup

The final setup has all the hardware components used in the project as shown in the Fig 9. The main component of the setup is the 3D printed ventilation system that is designed to provide ventilation to patients in an emergency situation. The system is controlled by a Raspberry Pi, which controls the ventilation rate using a stepper motor. The stepper motor is connected to a stepper motor driver, which amplifies the signal from the Raspberry Pi and provides the necessary power to the motor. The ventilation system is connected to an AMBU bag, which is used to provide the necessary air pressure to move the air in and out of the patient's lungs. The Raspberry Pi also connects to AWS IoT, which is used to store and analyze the data collected from the pressure sensor and stepper motor. The system is powered using power source from the plug. The final setup is a robust and effective solution for providing emergency ventilation to patients in critical situations. However, further testing and validation of the device are necessary before widespread adoption.

Fig. 9. Final Setup

**FUTURE WORK**

In the future, there is scope for further development and improvement of this emergency ventilation system. One area of focus could be to make the system more portable and easy to use in different settings. Currently, the system is designed to be used in a hospital setting, but with some modifications, it could be made suitable for use in other locations, such as ambulances, disaster sites, or remote areas. Moving forward, there are several areas of future work that can be explored based on the results of this project. One potential direction is to expand the dataset by collecting data from a larger sample size of patients with varying demographics and health conditions. This can improve the accuracy and generalizability of the dataset, allowing for more robust machine learning models to be developed. Additionally, incorporating real-time feedback and machine learning algorithms into the ventilation system can further optimize the performance and accuracy of the system. Another area of future work is to explore the potential of using other types of sensors and data sources to improve the ventilation system's accuracy and effectiveness. For example, incorporating data from pulse oximeters or other vital sign monitoring devices can provide additional insight into the patient's condition and allow for more precise adjustments to the ventilation system. Furthermore, integrating advanced machine learning techniques such as deep learning or reinforcement learning can lead to more sophisticated and adaptive systems that can continuously learn and improve over time. Overall, the potential for further research and innovation in this field is vast, and this project serves as a foundation for future work in the intersection of machine learning and emergency patient care.

## CONCLUSION

In conclusion, the development of a low-cost emergency ventilator using open-source hardware and software is a cru- cial step towards addressing the shortage of ventilators in underdeveloped regions of the world. The proposed system demonstrates that it is possible to design a functional and reli- able emergency ventilator using off-the-shelf components and a Raspberry Pi. The project can also be extended to support a range of features, such as remote monitoring and control, as well as data analytics and machine learning. Overall, the project highlights the potential of open-source technologies to facilitate the development of low-cost medical equipment, and underscores the importance of collaboration between various stakeholders in addressing global health challenges. By sharing the design and source code of the project, we hope to encourage others to build upon our work and improve access to emergency medical equipment for those in need. The COVID-19 pandemic has underscored the critical need for affordable and accessible healthcare equipment, and we believe that this project can contribute to addressing this challenge in the future. Further testing and validation of the device are necessary before widespread adoption.

## REFERENCES

[1]C. T. Truong, K. H. Huynh, V. T. Duong, H. H. Nguyen, L. A. Pham, and T. T. Nguyen, “Characteristic of paddle squeezing angle and ambu bag air volume in bag valve mask ventilator,” arXiv preprint arXiv:2109.08019, 2021.

[2]R. L. Kshetry, A. Gupta, S. Chattopadhyaya, M. Srivastava, S. Sharma, J. Singh, A. D. Gupta, and S. Rajkumar, “Design and analysis of a low-cost electronically controlled mobile ventilator, incorporating mechanized ambu bag, for patients during covid-19 pandemic,” Journal of Healthcare Engineering, vol. 2022, 2022.

[3]R. B. Kale and S. K. Singh, “Diy ventilator using arduino with blood oxygen sensing for covid pandemic,” 2022.

[4]A. M. Al Husseini, H. J. Lee, J. Negrete, S. Powelson, A. T. Servi, A. H. Slocum, and J. Saukkonen, “Design and prototyping of a low-cost portable mechanical ventilator,” Transactions of the ASME-W-Journal of Medical Devices, vol. 4, no. 2, p. 027514, 2010.

[5]O. Garmendia, M. A. Rodr ́ıguez-Lazaro, J. Otero, P. Phan, A. Stoyanova, A. T. Dinh-Xuan, D. Gozal, D. Navajas, J. M. Montserrat, and R. Farre ́, “Low-cost, easy-to-build noninvasive pressure support ventilator for under-resourced regions: open source hardware description, performance and feasibility testing,” European Respiratory Journal, vol. 55, no. 6, 2020.

[6]M. Shahid, “Prototyping of artificial respiration machine using ambu bag compression,” in 2019 international conference on electronics, information, and communication (ICEIC). IEEE, 2019, pp. 1–6.

[7]A. I. Abboudi, A. I. Alhammadi, K. M. Albastaki, A. Jarndal et al., “Design and implementation of portable emergency ventilator for covid- 19 patients,” in 2022 Advances in Science and Engineering Technology International Conferences (ASET). IEEE, 2022, pp. 1–4.

[8]L. Acho, A. N. Vargas, and G. Pujol-V a´zquez, “Low-cost, open-source mechanical ventilator with pulmonary monitoring for covid-19 patients,” in Actuators, vol. 9, no. 3. MDPI, 2020, p. 84.

[9]M. Robati, A. Sadighi, M. A. Nazari, and H. Naseri, “Momed: An affordable mechanical ventilator,” in 2022 30th International Conference on Electrical Engineering (ICEE). IEEE, 2022, pp. 345–350.

[10]P. Pelosi, L. Ball, C. S. Barbas, R. Bellomo, K. E. Burns, S. Einav, L. Gattinoni, J. G. Laffey, J. J. Marini, S. N. Myatra et al., “Personalized mechanical ventilation in acute respiratory distress syndrome,” Critical Care, vol. 25, no. 1, pp. 1–10, 2021.

[11] Rajender, G., et al. "Design and development of low-cost greenhouse to raise different cultivars." Int. J. Agric. Sci. Res. 7.3 (2017): 29-36.

[12] Marathe, V. I. N. A. Y. A. K., et al. "An evaluative effect of temperature on the human cognitive response using stroop test." International Journal of Mechanical and Production Engineering Research and Development 8.6 (2018): 75-80.

[13] Tuzon–Guarin, Joanna Marie. "Housekeeping Management Practices and Standards of Selected Hotels and Restaurants of Ilocos SUR, Philippines." International Journal of Business Management & Research (IJBMR) ISSN (P) (2016): 2249-6920.

[14] Dey, Abhirup, Rajasekaran Rajkumar, and Jolly Masih. "An analytical study and visualisation of human activity and content-based recommendation system by applying ml automation." International Journal of Mechanical and Production Engineering Research and Development 9.3 (2019): 75-88.

[15] Karishma, A., et al. "Smart office surveillance robot using face recognition." International Journal of Mechanical and Production Engineering Research and Development 8.3 (2018): 725-734.

[16] Miyamoto, Michiko. "Predicting Default for Japanese SMEs with Robust Logistic Regression." International Journal of Economics, Commerce and Research (IJECR) 6.3 (2016).