

Implementation of a Ventilator System for Efficient and Automated Respiratory Support using Arduino Controller

Dhanushree ML¹, Keerthana Mohan², Vamshi KL³, Prof. S S Vidya⁴

^{1,2,3}Student, Department of Electronics and Instrumentation, BIT, Bengaluru, Karnataka, India.

⁴Assistant Professor, Department of Electronics and Instrumentation Engineering, BIT, Bengaluru, Karnataka, India.

Abstract - The ventilator system is essential for addressing the shortage of medical ventilators, especially during emergencies, such as the COVID-19 pandemic, or in low-resource areas. It offers a low-cost, easily buildable alternative using Arduino and basic components, making it accessible and deployable in areas where conventional ventilators are unavailable. Compared to existing methods, it stands out for its affordability, simplicity, portability, and real-time control of key breathing parameters. Its open-source nature also encourages innovation and educational use, making it a valuable tool for both emergency response and learning. Utilizing an Arduino microcontroller and ESP8266 Wi-Fi module, the system automates a manual Ambu bag to deliver controlled ventilation. Key features include an adjustable breathing rate, tidal volume, and inspiration-expiration ratio, with real-time monitoring via an LCD display.

Key Words: Ventilator, Arduino Microcontroller, ESP8266 Wi-Fi Module, Real-Time Monitoring, Emergency Alerts, Ambu Bag, Controlled Ventilation.

1. INTRODUCTION

The human respiratory system functions through negative pressure generated by the diaphragm's movement, enabling inhalation and exhalation. During pandemics, the global shortage of ventilators severely strained healthcare systems, particularly in underprivileged regions with limited access to such critical equipment. In response to this crisis, some hospitals resorted to ventilator-sharing protocols, while temporarily addressing shortages, posed significant risks of cross-infection and improper ventilation. To mitigate these challenges, researchers have been actively developing low-cost, open-source ventilators to enhance accessibility and affordability. In addition to respiratory support, continuous monitoring of vital parameters such as heart rate remains essential for effective patient management, as it provides critical insights into a patient's physiological condition. A heartbeat can be measured manually by detecting the pulse at the wrist (radial pulse) or neck (carotid pulse) using fingertips. This ventilator system aims to design a low-cost and reliable ventilator system using Arduino microcontroller and ESP8266 Wi-Fi module. The ventilator uses a DC motor with a linear arm mechanism to automatically compress a silicon bag, providing breathing support for patients. To

monitor the patient's condition, sensors such as the MAX30102 are used to measure blood oxygen levels (SpO₂) and heart rate, while the DHT11 sensor records temperature and humidity. A 16x2 LCD display and control buttons allow users to start or stop the ventilator and choose suitable settings based on the patient's age group. Additionally, potentiometers are included to adjust the pressure and air volume delivered, ensuring the ventilator can be tailored to meet individual patient needs.

2. METHODOLOGY

The implementation of an Arduino-based ventilator system follows a structured methodology to ensure efficiency, reliability, and automation. The key steps involved are:

Step 1: Problem Analysis and Objective Definition

Identifying the need for an automated ventilator system. The key objectives are defined, focusing on maintaining optimal breathing cycles, pressure regulation, and adaptability to patient needs. Design constraints such as cost, power consumption, and component availability are also considered to ensure feasibility.

Step 2: System Design

Selecting appropriate hardware and software components. Hardware selection includes an Arduino microcontroller, ESP8266 Wi-Fi module, Ambu Bag, DC motor, LCD display, and a power supply. Software logic is developed using the Arduino IDE, focusing on sensor data acquisition, real-time processing, and adaptive breathing cycle control.

Step 3: Prototype Development

All selected components are assembled into a functional system. The Arduino program is written and uploaded to control air delivery based on sensor inputs based on various age groups.

Step 4: Software Development

Programmed Arduino with a breathing cycle control algorithm (inhale–hold–exhale–pause). Integrated sensor data acquisition and real-time monitoring. Used an app called

ThingShow for receiving real time data on mobile phones and pc.

Step 5: Testing and Optimization

Ensuring the ventilator operates efficiently and safely. Initial testing is conducted using a controlled model or test lung to validate airflow rates, pressure regulation, and response time. Software algorithms are optimized to enhance efficiency and reduce latency.

Step 6: Deployment and Evaluation

The ventilator is deployed in real-world settings such as hospitals, emergency care, or home use. System performance and reliability are continuously monitored to identify potential improvements. Data is collected for further refinements, and user guidelines are provided to ensure proper operation.

3. BLOCK DIAGRAM

The block diagram of the Ventilator System illustrates the working principle of an IoT-based smart ventilator system. The core of the system is the Arduino microcontroller, which serves as the central processing unit that coordinates all components. It receives input signals from various sensors such as the heartbeat sensor, temperature sensor, and blood oxygen sensor (SpO₂ sensor) to continuously monitor the patient's vital parameters. The microcontroller processes this data and displays real-time readings on the LCD display and an app called ThingShow for easy observation. The ventilator unit, controlled via a driver circuit, provides automated airflow to the patient, with purified air supplied from an Ambu Bag. The system is powered by a regulated power supply to ensure stable operation. For connectivity, the IoT modem transmits the collected data to the cloud, enabling remote monitoring of patient health by medical personnel. This integrated setup makes the ventilator intelligent, efficient, and suitable for real-time patient monitoring in both hospitals and remote healthcare environments.

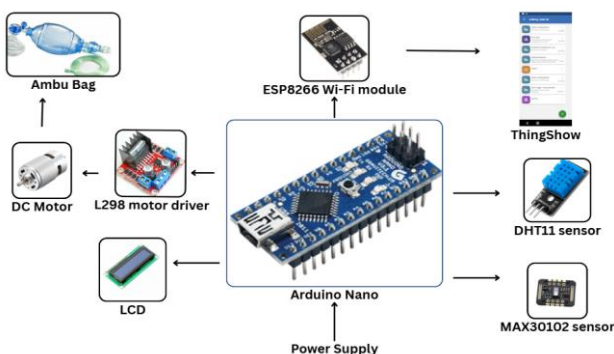


Fig -1: Block Diagram

4. WORKING

The System begins with the initialization of components, including the Arduino microcontroller, sensors like DHT11 and MAX30102, LCD display, and DC motor. Once initialized, the system reads vital parameters such as temperature, humidity, and beats per minute (BPM) from the sensors and displays them on the LCD. The user then sets the appropriate age group (Adult, Child, or Infant), to adjust the ventilator's operation based on the patient's breathing requirements. The system continuously checks the measured parameters against critical thresholds to ensure patient safety. If the readings are within normal limits, the ventilator maintains its current operation. However, if abnormal values are detected, the system adjusts the airflow using a potentiometer (pot) to stabilize the patient's condition. Finally, the system sends all monitored data to the cloud platform (ThingSpeak or ThingShow) for remote monitoring and data analysis before stopping or looping back for continuous operation. This flow ensures real-time monitoring, adaptive control, and IoT integration for enhanced healthcare support.

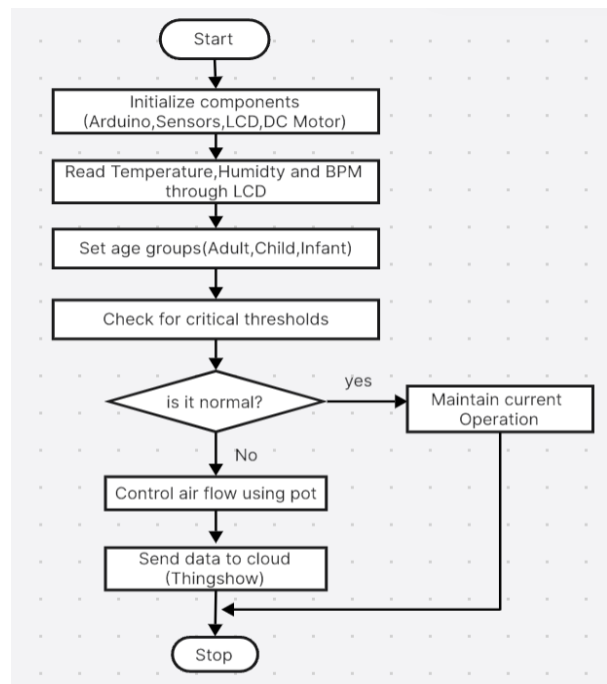


Fig -2: Flow Chart

5. DESIGN AND CIRCUIT DIAGRAM

The circuit diagram shows an IoT-based Arduino ventilator system built around an Arduino Nano. It connects a DHT11 sensor for temperature and humidity, and a MAX30102 sensor for blood oxygen and heart rate monitoring. A 16x2 LCD with I2C module displays real-time data. The L298N motor driver controls a DC motor that operates the ventilator mechanism, with potentiometers used to adjust airflow. Push buttons allow manual input, and an ESP8266 Wi-Fi module sends data to the cloud for remote monitoring. The setup is

powered by a regulated supply, enabling a compact, low-cost, and efficient ventilator control system.

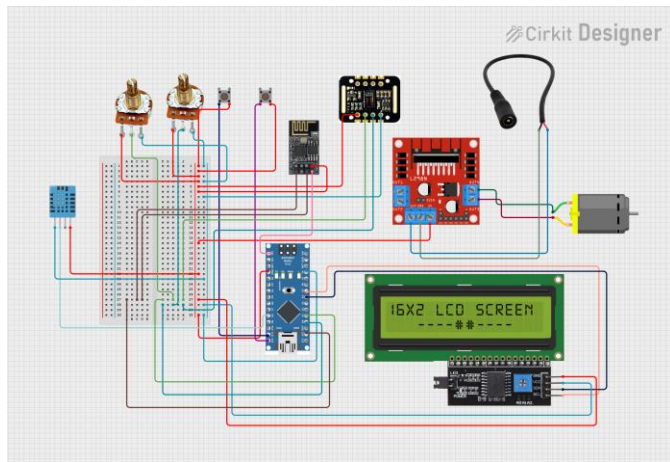


Fig -3: Circuit Diagram

6. RESULTS AND DISCUSSION

The implementation of the IoT-Based Smart Ventilator System has led to major improvements in affordable healthcare automation and patient monitoring. The key outcomes observed include:

1. **Automated Ventilation Control:** The system efficiently operates a DC motor mechanism to compress the ventilator bag, providing smooth and consistent airflow according to the patient's needs.
2. **Real-Time Health Monitoring:** Integrated sensors such as MAX30102 and DHT11 continuously monitor vital parameters like SpO₂, heart rate, temperature, and humidity, displaying accurate readings on the LCD.
3. **IoT-Based Data Transmission:** Using the ESP8266 Wi-Fi module, the ventilator successfully transmits real-time health data to cloud platforms like ThingSpeak, enabling remote monitoring and record-keeping.
4. **Automatic Alerts and Adjustments:** The system detects abnormal readings and automatically adjusts airflow while triggering alerts, ensuring patient safety and timely intervention.
5. **Enhanced Accessibility and Affordability:** The low-cost, portable design makes the ventilator suitable for remote healthcare centers, emergency use, and resource-limited areas, contributing to more accessible life-support solutions.

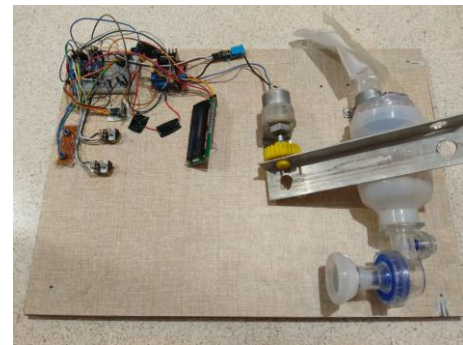


Fig -4: Ventilator System model

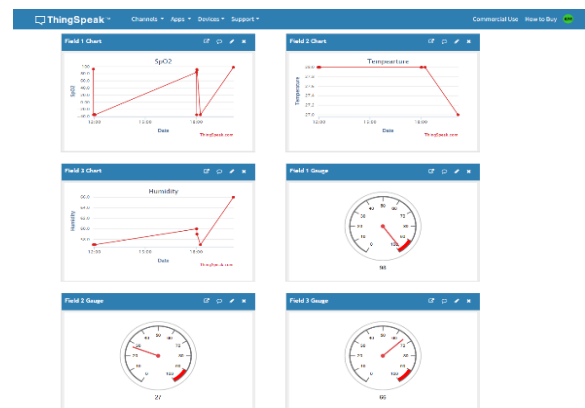


Fig -5: ThingSpeak

7. CONCLUSION

The implementation of an Arduino-based ventilator system demonstrates that affordable, efficient, and automated respiratory support can be achieved using embedded technology. The prototype successfully regulated airflow, monitored patient parameters, and reduced manual intervention, proving its potential as a reliable solution in emergencies and resource-limited healthcare settings. This Ventilator System highlights how low-cost automation and sensor integration can address critical medical challenges. With further improvements in portability, IoT integration, and compliance with medical standards, the system can evolve into a practical, scalable solution for hospitals, rural health centers, and disaster management scenarios.

REFERENCES

- [1] S. Rajalakshmi, S. Kavipriyabai, S. Vinodhini and S. Krithika, "Ventilator Using Arduino with Blood Oxygen Sensor," 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2022, pp. 01-05, DOI: 10.1109/IC3IOT53935.2022.9767888.
- [2] H. Gaur, S. Yadav, Shashikant, A. Shahin and M. Shahid, "Review On Design and Performance of the Medical Ventilators," 2023 International Conference on Power,

- Instrumentation, Energy and Control (PIECON), Aligarh, India, 2023, pp. 1-5, DOI: 10.1109/PIECON56912.2023.10085815.
- [3] R. B. Mohite, U. P. Waghmode, A. J. Chinchwade, S. D. Jadhav and V. E. Pawar, "E ventilator (Emergency ventilator) for Emergency Situations," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-6, DOI: 10.1109/I2CT57861.2023.10126249.
- [4] https://www.researchgate.net/publication/370989811_E-ventilator_Emergency_ventilator_for_Emergency_Situations.
- [5] https://www.researchgate.net/publication/392076594_Patient_Monitoring_Low_-_Cost_Automated_Ventilator.
- [6] V. Krishna Prasath, R. Kiruthika Uma, R. Deekshita, T. Arun Prasath and M. Pallikonda Rajasekaran, "Smart Automated Emergency Ventilator (SAEvent): Design of a Low-Cost, Innovative And Portable Ventilator," 2023 9th International Conference on Smart Computing and Communications (ICSCC), Kochi, Kerala, India, 2023, pp. 606-609, doi: 10.1109/ICSCC59169.2023.10335041.
- [7] https://youtu.be/bS_2QTzhkDY.
- [8] A. R. da Silva et al., "Low-Cost Open-Source Ventilator Prototypes for COVID-19," IEEE Access, vol. 9, pp. 122-132, 2021, doi: 10.1109/ACCESS.2021.3050799.
- [9] S. P. Lal and P. S. Velmurugan, "Design and Development of an IoT-Based Smart Ventilator," 2021 International Conference on Smart Electronics and Communication (ICOSEC), pp. 130-135, doi: 10.1109/ICOSEC51865.2021.9591715.
- [10] A. Sharma et al., "Development of a Low-Cost Mechanical Ventilator Using Arduino," 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1-6, doi: 10.1109/CONECCT50063.2020.9198535.