Title: Rapidly developable low cost and power efficient portable turbine-based emergency ventilator

Authors: Syed Razwanul Haque[a], A K M Maruf Hossain[a], Shovon Sudan Saha[b], Md Hasanur Rahman Sohag[a], Fozle Rabbi Shafi[a], Satya Ranjan Sarkar[c]

Affiliations:
[a]CRUX Technologies, Sylhet, Bangladesh
[b]Bioforge Health Systems Limited, Dhaka, Bangladesh
[c]Oishi Electronics, Sylhet, Bangladesh

Contact email: nabil@cruxbd.com

Abstract: Use of ventilators has always been common in medical scenarios but very expensive to procure or develop; mainly because of expensive components and precise instrumentation. Our paper attempts to mitigate that problem by proposing a novel way to rapidly develop a portable ventilator that uses 3D printing technology and off-the-shelf components. This turbine and valve-based ventilator features most commonly used modes. A unique servo-based pressure release mechanism has been designed that makes the system around 36 times more efficient than solenoid-based systems. Reliability and efficiency have been increased further through the absence of electromechanical components in our novel positive end expiratory pressure (PEEP) valve. Effective algorithms such as feed-forward and PID are used alongside our unique 'Sensor data filtration Methodology'. It provides an interactive GUI through an app that can be installed on any readily found tablets while the firmware manages the breathing detection algorithm using the flow sensor. This modular, portable, energy-efficient and low-noise system ventilator also features a swappable battery that can run for 5.6 hours at a stretch and also holds the ability to even run-on solar power. This ventilator's design and development files system have been open-sourced and can be found at: (https://www.github.com/nabilphysics/ventilator)

Keywords: *turbine-based ventilator, sfm3300, mpx2010, prvc, simv, peep valve, 3d printing, covid-19, pressure release mechanism, 18650 battery, acute respiratory distress syndrome, medical, emergency, ventilator, turbine-based, low cost, modular, portable, peep*

1. Hardware in context

Medical Ventilator is one of the most integral part of an ICU setup, however, its significance became very apparent to the mass people when Covid-19 struck the world and cases with acute respiratory distress syndrome (ARDS) condition amplified. It was not long after when the crisis for ventilators strikes, the manufacturing companies ramped up their production but it was just not enough. Due to its inherited complexity in both the development process and supply chain, not much could have been possible. Besides that, let alone the cost of the ventilators, even the procurement process is not straightforward. Nevertheless, most of the devices that are found on the market have proprietary parts and complicated operations that require skilled personnel.

2. Hardware description.

A low-cost portable emergency ventilator which is rapidly developable using off-the-shelf components. It holds most of the modes that are commonly used by doctors. Some of the advantages of using our system are as follows:

- Low-cost and robust: Development is cost is very less as common technology such as fused deposition modeling (FDM) technology; 3D printing, used and most of the components used are very much available on the market and are also used in other devices such as drones, other respiratory devices, etc.
- Portable with long battery backup: While ventilators are usually bulky, over the last years, there have been developments of portable ventilators too, however, they are still very bulky with low battery life. On the other hand, our device weighs only 3.7kg and runs for 5.6 hrs. at a go. Rated at around only 16W, our device can run for a long time. Swappable and modular battery systems enable the system's operation time to be increased for extended periods of time. Moreover, space has been kept for anyone willing to add more batteries.

  Often patients are one the move, let it be within hospitals, from hospital to another or in an ambulance from an accident scenario, ventilators are usually a necessary device necessary for critical care patients, but most of the available ventilators are not fit for these scenarios. Here's where our device's portability and long battery backup would help alleviate the situation.

  Besides that, many remote areas are not connected to the power grid of the country; especially in lower- and middle-income countries (LMIC), and this acts as a huge barrier to using biomedical devices in the clinics and hospitals that are barely available. To lessen such problems our device can be charged using solar energy, wind energy, etc. where reliable constant power source is not available. Nevertheless, it can also be used as a CPAP and BiPAP device at home.

  High efficiency and low noise (26.1 dB from 1m distance) has been achieved through the novel design of our pressure release mechanism (PRM) and positive-end-expiratory-pressure (PEEP) valves. A servo-based mechanism has been used that can easily replace the power-hungry solenoid-based mechanism that are to be found in any ventilators. More than 36 times more efficiency can be achieved using such a system. This comes to a great advantage along with portability.

  This mechanism can also be used in various pneumatic systems such as compressors, safety valves, oxygen concentrator and fuel control system.
- Adaptable to any patient circuit: Upon market analysis, it has been observed that all of the ventilators come with their proprietary patient circuits and in many scenarios if this part of the system fails, the entire ventilator system becomes unusable. To mitigate this problem, our system is adaptable to any patient circuits that are available on the market.
- Graphical User Interface (GUI) is adaptable to any android device: An android application is developed that communicates with the system through a predetermined data frame using Bluetooth communication channel. The low size and efficient software architecture enable the app to be highly responsive and require small storage space, subsequently empowering the app to be installed in just about any device with an android operating system.

Nevertheless, it can also be connected to single-board computers such as: Raspberry Pi [1], Radxa Zero [2].

- Ability to power itself using solar panels and other alternate resources: Still in a lot of LMICs, there are many regions where power through the grid has not been able to reach, unfortunately most of those regions are seen to be struck with natural disasters; a scenario where logistically it is not possible to bring out the patients to provide critical by taking elsewhere nor it is possible to provide treatment there because of the lack of infrastructures for critical medical devices such as ventilators.

  Our device's low power requirement and large range of voltage; both AC and DC, allows it to run on even renewable resources such as solar, wind, tidal energy, etc. Making the device typical to be deployed in low resource settings and disaster-stricken scenarios where there is not even proper power available from the grid. Nevertheless, it can also be used in refugee-stricken areas.

To provide the advantages, most parts of the system can be developed using 3D printer and some commonly found components. The entire flow of the system is described in [Figure 1].

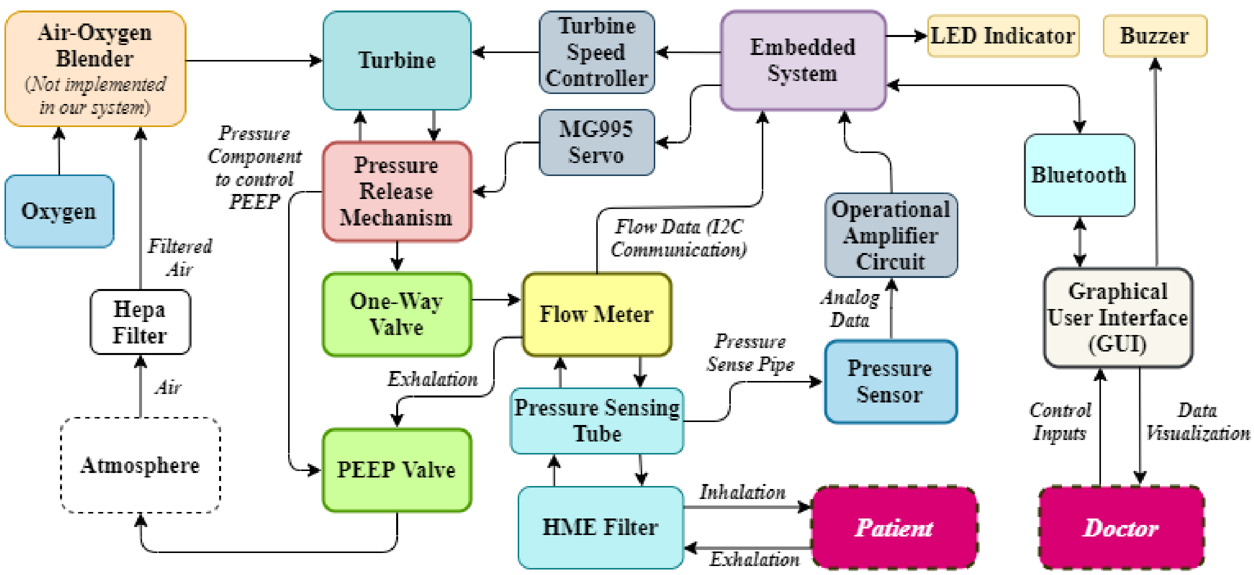

Figure 1: System Architecture for our turbine-based ventilator

The system architecture [Figure 1] portrays the air flow from the atmosphere to the patient, while passing through critical components of our system. Firstly, the air is filtered through the HEPA filter, a component that is well known for its structure as it holds the capability to filter particles ranging from 10 to 0.01 microns according to [3]; enough to stop Covid-19 particles as they have an approximate diameter minimum to 0.06 microns [4]. The filter air then enters the turbine which is being controlled by the embedded system through the turbine speed controller. Afterwards, the air travels to PRM to one-way valve and finally to the patient after passing through the flow meter, pressure sensing tube and HME filter. Bi-directional filtration is done through the HME filter [5] to filter any possible virus particles from the patients' breath in order to restrict spreading of viruses when they are eventually released into the atmosphere through the peep valve.

This architecture also explains how data is being collected and amplified before feeding into the embedded system. The embedded system is also responsible to send and fetch data using Bluetooth from the GUI where doctors' inputs settings for the ventilator. Besides that, also triggering any alarms if necessary.

2.1 Mechanical

a. **Turbine:** Because of its capacity to deliver high airflow, low noise, minimal temperature increases during operation, and small size, the 7040 DC 12V centrifugal turbo turbine [6] was chosen. It also features a built-in control circuit that makes regulating the entire system's air flow considerably easier. Finally, the ventilator's three-phase brushless motor allowed it to function at an extremely low power rating

b. **Pressure Release Mechanism (PRM):** We need to quickly raise and reduce pressure during inhale and exhale, therefore a servo-actuated system employing the MG995 (explained more in the 'Servo' section), which runs on 5V and has a stall torque of 9.79kg/cm [7], was created to do just that. One of the measures we took was to keep the turbine's speed as low as possible so that it didn't have to start from zero, which would have resulted in increased inertia. It is therefore simple to raise the pressure fast to the desired pressure. The Pressure Release Mechanism is also responsible for diverting the pressure while the turbine is running and controlling the Positive End-Expiratory Pressure (PEEP) valve synchronization. Section analysis of the PRM is being shown where all the parts are visible [Figure 2].

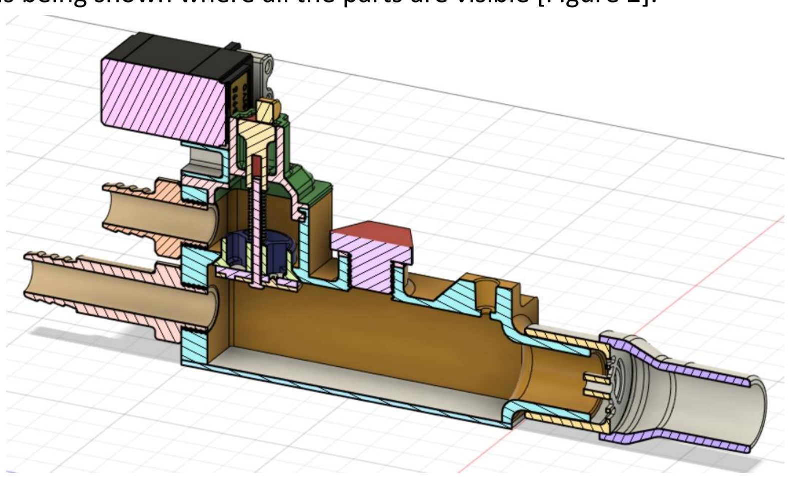

Figure 2: Cross-section of the PRM

In the [Figure 3], the PRM is in the inhale state. The manifold, valve and guide of the PRM are all to be 3D printed to be easily deployable and fit into the system. The guide is glued onto the manifold and it provides a supported pathway for the valve to move. A spring is installed in order to keep the valve pushed upwards so that during inhale state, air can flow from the inlet directly with high pressure (P1); ultimately providing P2 at the outlet. Pressure (P3) is zero here, hence, at one of the other outlets of PRM. Magnitude of the pressures depend on the [Equation 1]. P2 and P4 will always be the same as they are the outlets of the same manifold.

$$P1 = P2 + P3 + P4 \text{ --- } (1)$$

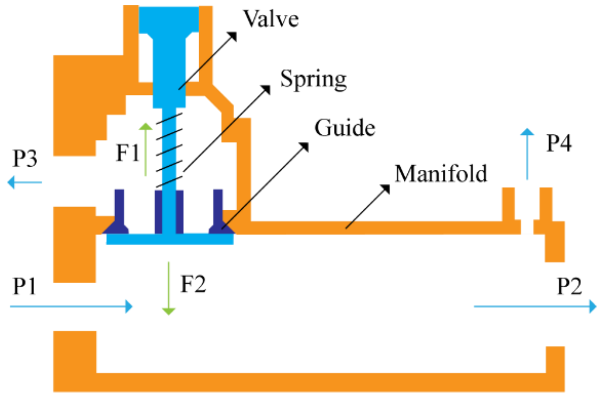

Figure 3: Free-body diagram of PRM during inhale

During the exhale state [Figure 4], the horn of the servo is rotated so that the valve is pushed downwards, compressing the spring and opening up the space in between the guide. A large of portion of the pressure now deviates through that gap and rising the pressure P3 and decreasing the pressure P2. P2 pressure plays a vital role in the system to keep a positive PEEP to restrict the alveoli from collapsing. When the system enters the inhale state, the servo again retains its original position and the valve again closes itself due to the force (F1).

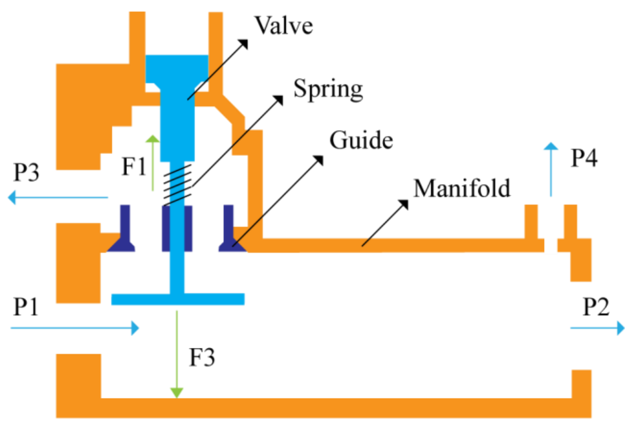

Figure 4: Free-body diagram of PRM during exhale state

To summarize, this novel design of the pressure release mechanism has the ability to operate at very low pressures such as 0 cm to 95 cm H_2O (tested) during the exhale state, minimize failure as it is normally open and power efficient. The air travels from the turbine through this mechanism and then reaches the patient after passing the 'one-way valve'.

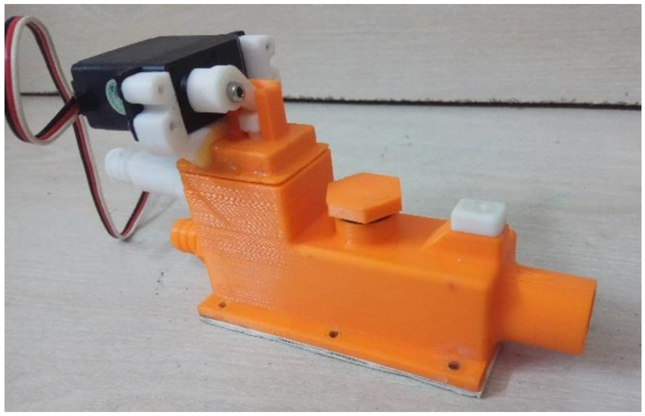

Figure 5: Pressure Release Mechanism with MG995

c. **One-way valve:** was custom designed by our team to fit as the market available products were not much available and procuring such a valve would have taken a long time. A number of iterations were experimented and finally a design [Figure 6 and 7] was finalized that worked brilliantly with 5cm H_2O. This 3D printed structure holds a rubber membrane on the inside and can fit any readily available patient circuit.

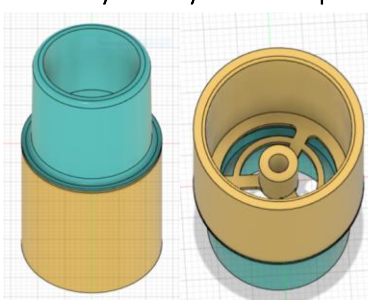

Figure 6: One way valve

Figure 7: One way valve (real-life)

d. **Positive End-Expiratory Pressure (PEEP) valve:** Maintaining the PEEP value ranging from 5 to 20 cm H_2O of patients is highly essential so that the Alveoli of the patients do not collapse and there is some volume present inside, which makes it easier to inflate; this point of change in compliance that is denoted as the critical opening pressure (COP) [8]. PEEP is regulated by doctors depending on the patients' lung compliance.

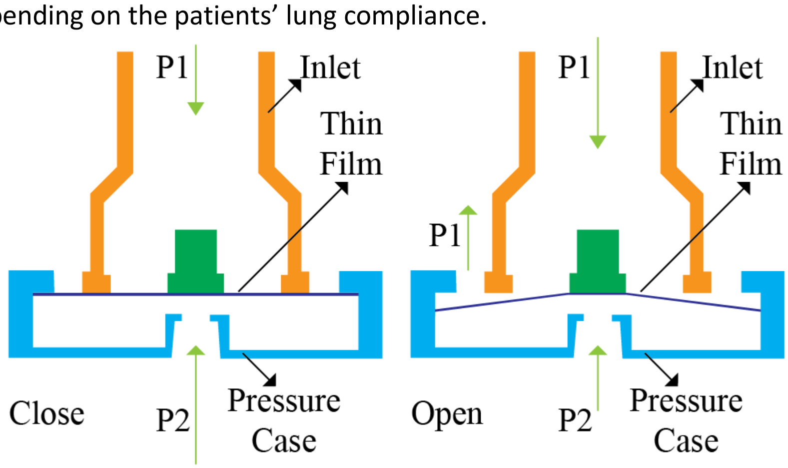

Figure 8: Free-body diagram of the PEEP valve

[Figure 8] shows the free-body diagram of the PEEP valve during both the inhale and exhale states to the left and right respectively. Both the pressure case and inlets are 3D printed using PLA and a thin film made up of silicone sheet is added in-between of both the parts. During the inhale state, a high pressure is exerted (P2) through the pressure component taken from the PRM and the thin film is pushed tightly towards the inlet as there is a low pressure (P1) coming from the patient. However, quite the opposite happens when the patient is in an exhale state and the pressure P1 now is much higher than that of P2, leading to pushing down the film and letting the air out.

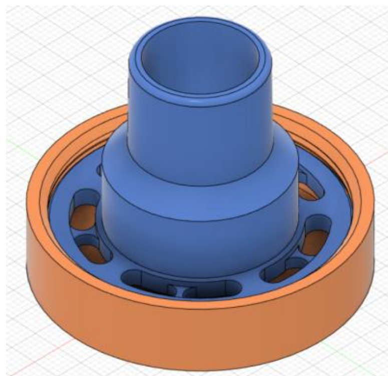

Figure 9: Tailored PEEP valve

This novel design and mechanism of PEEP valve [Figure 9] operates without the aid of any electromechanical components and this enhances the reliability of the entire system. In the scenario of failure, the valve would just open and avoid the suffocation of the patient. Moreover, designing the control algorithm becomes also easier as the only variable one needs to take care of is only the speed of the motor.

Finally, this decreases the time taken for assembly [Figure 10], complicating of supply chain due to lower number of parts and eventually reducing the cost of production.

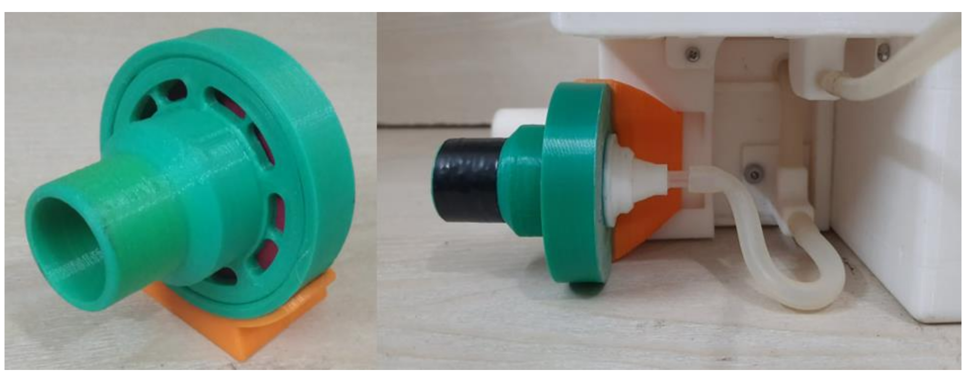

Figure 10: PEEP valve after assembly

2.2 <u>Electrical and electronics:</u>

To run the mechanical systems seamlessly a robust circuit is essential that will be integrated with the electronic modules [Figure 11]. Generic modules are chosen in the development so that it is abundantly available in the market which would eventually make the development process and supply chain much less complicated when it enters the production phase.

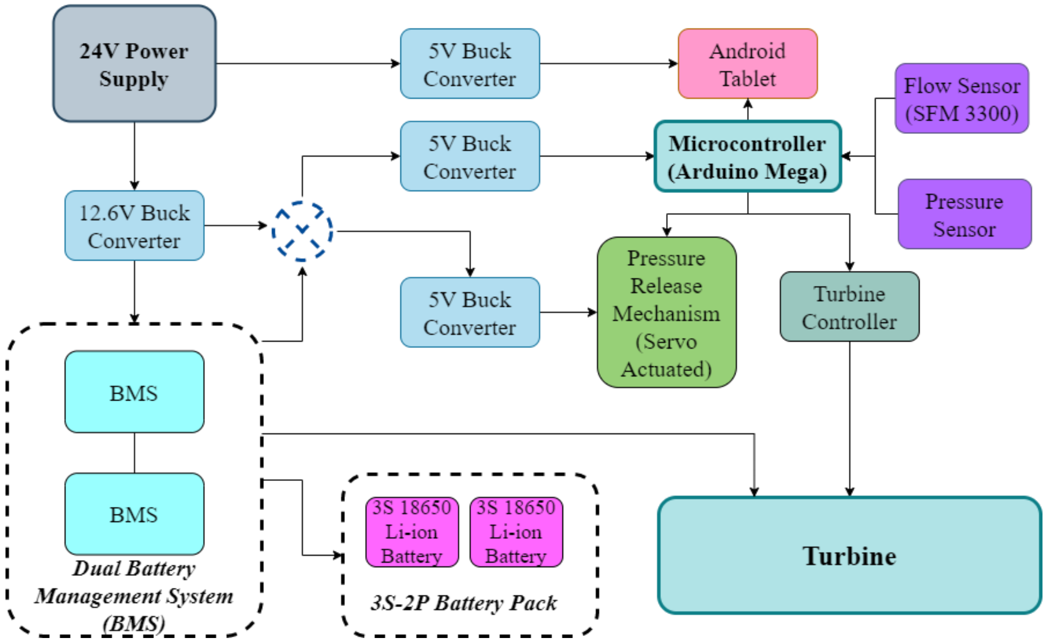

Figure 11: Electronics Modules' Integration

a. Key modules

i. Microcontroller:

Arduino mega [9] is being used as the processing unit of the device for its wide availability, robustness and numbers of digital pins available. It acts as the brain of the system taking in data from all of the sensors that are connected and producing appropriate control signals in order to control the turbine and servo. Besides that, data transmission through Bluetooth is also conducted by this microcontroller [Figure 12].

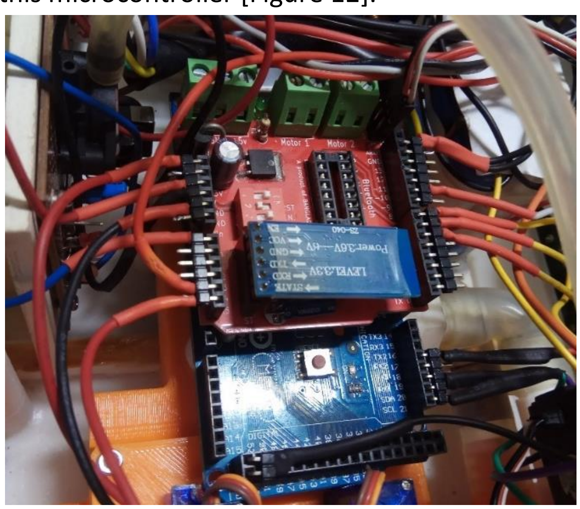

Figure 12: Embedded system

ii. Bluetooth module:
The user inputs are sent from the Android application to the microcontroller via a Bluetooth module (HC-05) [10] [Figure 13], which are then inputted into different control algorithms and other processes. Our app handled the serial communication seamlessly without any failure at a baud rate of 9600 where the predetermined data frame comprising of multi-variant values were lined up in single strings. Afterwards that string is segmented based on the preset data length of each variable and portrayed on the necessary graphs which had the capability to optimize its scale and provided better visibility to perceive useful information. One can also directly connect the device with our ventilator through the USB port.

Figure 13: HC-05 Bluetooth Module [7]

iii. Servo: MG995 servo [7] is an integral part of the PRM which helps to increase and decrease pressure in such a way that the synchronization in between the patients' breathing is present. This servo is specifically used because of its torque being 9.76kg/cm; enough to actuate the valve while being durable. Its low energy consumption is another major reason behind choosing the servo.

One of the key attributes for our system to be efficient is due to our servo-based pressure release mechanism's low power requirement. Upon experimentation on the certain conditions; explained in [Table 1], we found that in 50% duty cycle mode and continuous mode the energy consumption of our servo-based system is 19.84 times and 36.46 times more efficient.

Table 1: Power comparison of Pressure Release Mechanisms

Type	**Solenoid Based Pressure Release Mechanism**		**Servo Based Pressure Release Mechanism**	
Test Duration	5 minutes		5 minutes	
On/Off Duration	0.25 Hz (50% Duty Cycle)	Continuous On	0.25 Hz (50% Duty Cycle)	Continuous On
BPM	15	-	15	-
I:E	1:1	-	1:1	-
Average Energy Consumption (Wh)	1.042	1.611	0.050	0.043

iv. Turbine controller: The previously mentioned turbine comes with an integral turbine controller that can easily be used to control the PWM signals from the micro-controller. This controller removes all the complexity that are often present in controlling a motor/turbine. Hence, easy integration of the control algorithms is possible.

b. Sensors

i. MPX2010DP: is a differential pressure sensor [11] that takes in data from the pressure sense pipe mentioned in [Figure 1] and feeds analog data into the Arduino for the firmware to work. However, the sensor cannot be directly used to send the data to the microcontroller. The voltage that is normally generated from the analog output of the sensor is in millivolts and it shall be amplified in order for our microcontroller to pick up the signal.

An operational amplifier (LM358) is used for sensor signal amplification. [Figure 14] demonstrates the circuit used. It uses a very generic yet effective circuit.

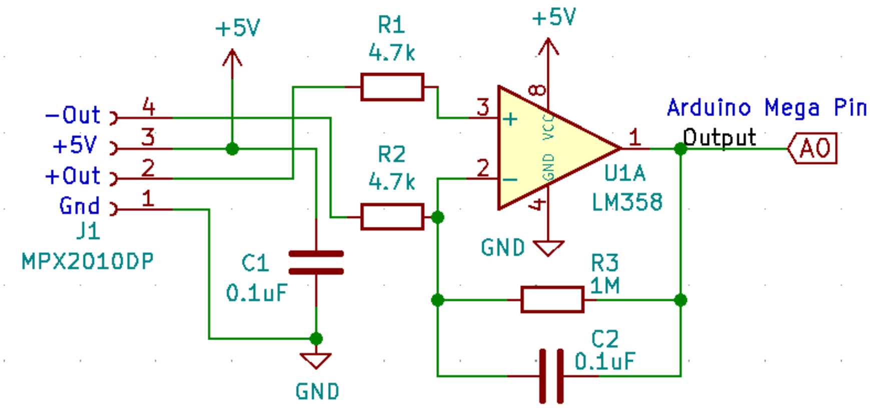

Figure 14: Amplifier (LM358) circuit schematic for MPX2010DP pressure sensor

ii. SFM 3300: is a flow sensor [Figure 16] works like a charm since it does not require any calibration and library to run the sensor is open source but an external circuit [Figure 15] is needed to have high precision data. On top of that, SFM3300 [12] draws less than 50mW of power; draining only 10mA current at 5V, enabling it to be powered by the microcontroller's digital pin only.

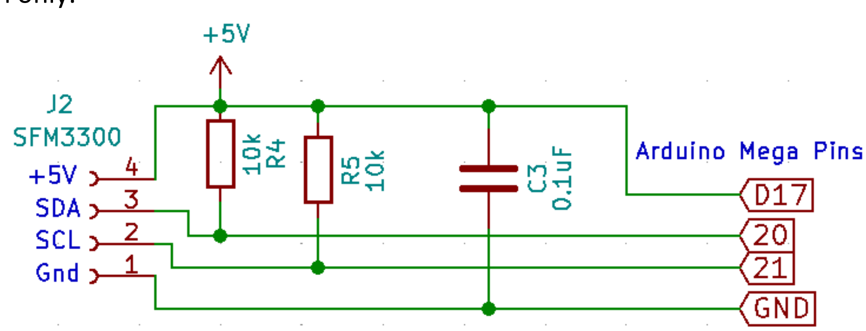

Figure 15: Circuit for SFM3300 with pull-up resistors

Figure 16: SFM 3300 flow sensor with custom 3D printed casing

b. Power modules: A modular design that is based on electronic modules that can easily be picked up from the market are used here. A Vero-board design [Figure 17] based on [Figure 11] is used so that it can also eliminate the dependency of manufacturing PCBs. But the design can surely be scaled to be manufactured using PCBs. Moreover, the use of electronic modules instead of individual SMD or THT components has enabled our system to be manufactured even in the situation where expensive and complex pick and place machines is not available.

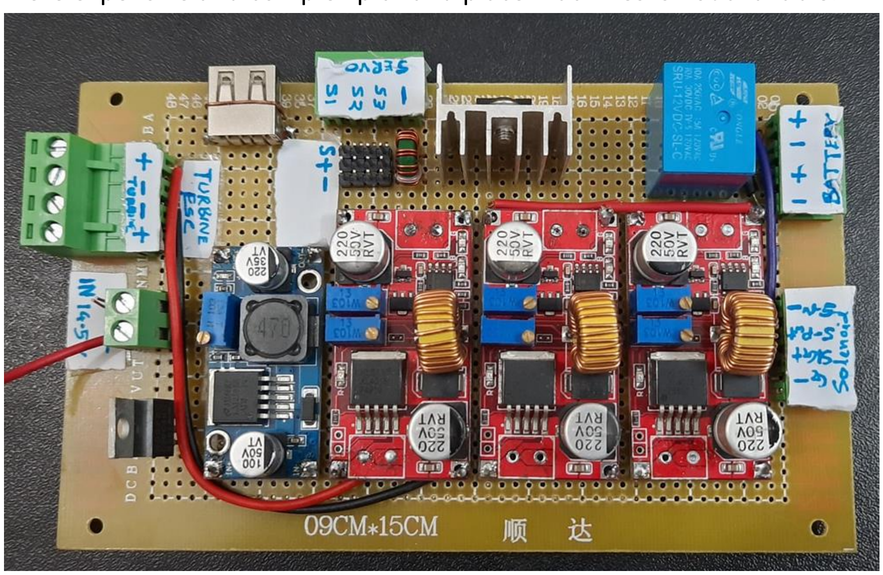

Figure 17: Power Distribution Board

i. Buck Converters: Three independent buck converters are used to provide power to three main modules of the system:
   a. Android Tablet
   b. Microcontroller
   c. Pressure Release Mechanism

ii. BMS: Two BMS modules are connected that work independently with two battery packs so that even if one of the BMS modules fails, there is still another as backup, hence increasing reliability further.

iii. Battery packs: Panasonic NCR18650B batteries [13] have been used as backup power instances of portable operation and during power outage/instability. A 3S2P configuration is used, ultimately getting 6800mAh, 12.6V and 85.68Wh. Our system is expected to run for 5.6 hours at a stretch with settings mentioned in [Table 2]. However, using lower pressure settings will enable the device to run even longer, for example, it can run for maximum 8 hours under 20 cm H_2O pressure setting.

Table 2: Ventilator system test summary

Test Duration (mins)	10
BPM	10
Inspiration Time, Ti (s)	2
Pressure (cm H_2O)	35
Energy Consumed (Wh)	2.544
Average Power Consumption (W)	15.27

2.3 Modes of Ventilation

Intensive Care Units (ICU) admitted patients are often given mechanical ventilation to reduce Work Of Breathing (WOB), improve oxygenation, or correct respiratory acidosis without damaging lungs while enabling the respiratory muscles to rest [14].

After intubating patients to ventilators, this objective can be achieved by doctors by many modes of ventilation. Considering the latest and most prominent modes, we have developed our system with the following modes:

a. Pressure Regulated Volume Control (PRVC) mode- is a combined volume and pressure regulated mode of ventilation that attempts to reach the targeted (user inputted) tidal volume (V_T) with the lowest possible pressure.

   PRVC helps prevent volutrauma and barotrauma [15] by limiting the delivery pressure as it is an adaptive control form of ventilation that permits involuntary tuning of targets (pressure versus volume) through the compilation of data from the previous breathing cycles. Flow rates are adjusted to reach patients' demands even when their lung compliance changes.

   PRVC adjusts flow rates to adapt with demand when lung compliance changes; subsequently achieving normal breathing pattern. This mode is often the go-to mode for infants with ARDS. Besides that, PRVC provides the comfort of pressure ventilation for patients with reliable minute ventilation.

b. PCV (Pressure Control Ventilation)- It's a pressure-controlled ventilation where the desired Peak Inspiration Pressure (PIP) is taken in from the doctor and the required tidal volume (V_T) is delivered in order to achieve that pressure. Besides that, the PEEP is also maintained as the minimum pressure [Figure 18]. Mandatory breathes are provided to the patients but if breath can be initiated by the patient if required; intermittent breathes [16]. However, considering patients' safety, a maximum pressure (P_{Max}) is hardcoded into the system.

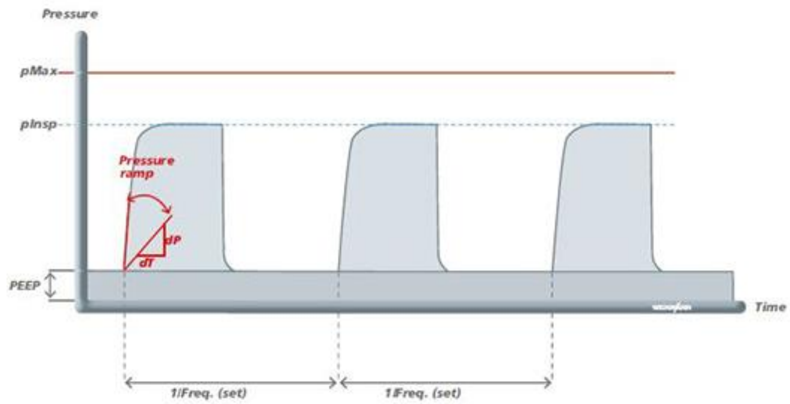

Figure 18: Pressure changes in PCV mode [14]

c. SIMV (Synchronous Intermittent Mandatory Ventilation)- "Synchronized" means that the ventilator will adjust the timing of the breaths and delivery in accordance with the patient's efforts. "Intermittent" means that only some of the breaths are supported, while "mandatory ventilation" a set rate; Respiratory Rate (RR), is selected by the doctor and the ventilator will deliver these minimum breaths every minute irrespective of the patient's effort of breathing. The mandatory breaths can be triggered by the patient or by time if the patient's RR is slower than the ventilator RR (as with AC). This mode is generally used as a technique of "training" the diaphragm in order to maintain the muscular tone and wean off patients from the ventilator in a quicker manner and finally, lower patients' dependency on ventilators [17].

   In our system we have kept SIMV as an additional feature that can be turned on in both PRVC and PCV modes of ventilation where patients would trigger intermittent breaths with a change in pressure and flow respectively.

2.4 Software

a) Android Application – To make the ventilator even more robust, an app was developed on the Android 9 weighing only 5 Mb of space and could seamlessly run on a tablet comprising of only 2 GB RAM and 16 GB ROM. The efficient development of the app made it possible to run using low resources, subsequently decreasing the price of the device needed to operate.

Amongst all the options, Java was chosen as the language to be used to develop because it is reliable, has many available libraries and support and easy adaptability to changes. Besides that, debugging is also expected to be easier due the development being done on Android platform on Java code base. Our app handled the serial communication seamlessly without any failure at a baud rate of 9600 where the predetermined data frame containing of multi-variant values were lined up in single strings. Afterwards that string is segmented based on the preset data length of each variable and portrayed on the necessary graphs which had the capability to optimize its scale and provided better visibility to perceive useful information.

An implementation of the multithreading [18] was inevitable because of our application's asynchronous [19] nature, which enabled us to maintain uninterrupted serial communication between the hardware and the software while performing background tasks. Firstly, software needs to establish communication with hardware and persist its connection for further data transmission. Secondly, there are many modes of the system like 'PRVC', 'PCV' and 'SIMV' that the software must be able to change in the runtime. Besides, the system must support initiating alarm in critical situations and snoozing it manually for example, when the value of 'Tidal Volume', 'PiP', or 'PEEP' is above or below the optimum limit.

To facilitate all the asynchronous activities, the 'Thread Looper Handler Architecture' [20] of android was put into use. When the software creates a socket [21] connection with the hardware, the software always receives and sends data in the form of a stream. Subsequently, a thread is assigned to manipulate the stream as the data in the steam is always in ASCII form. In, android it is not ideal to make the UI thread busy for performing heavy tasks as the UI might become unresponsive. So, we incorporate a handler [20] that can efficiently send data to the UI thread as well as receive it. The handler allows sending and processing Message and Runnable objects associated with a thread's MessageQueue. It will deliver messages and runnable to that message queue and execute them as they come out of the message queue.

User Interface (UI) and User Experience (UX) were given special attention and were developed while taking abundant inputs from doctors, nurses, etc. The interface is designed in a way that doctors can easily observe the critical data from a considerable distance at a glance and can conduct an operation with minimum clicks. Here, not only the alarms can only be seen clearly but also the interactive UI also permits the users to change the conditions given for the alarms; both upper and lower bounds, providing more confidence to run the ventilator.

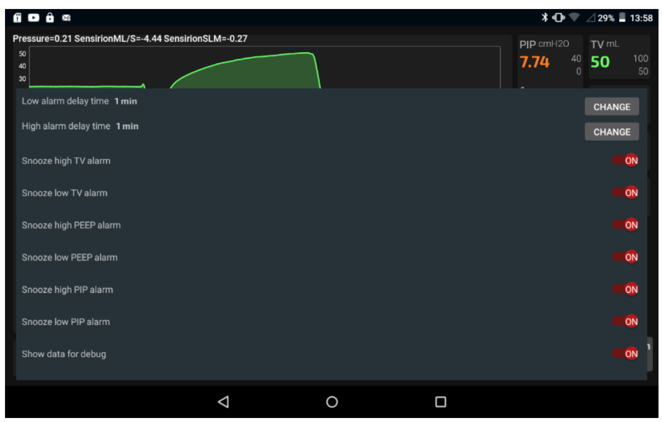

Figure 19: Alarms can be turned on/off from here

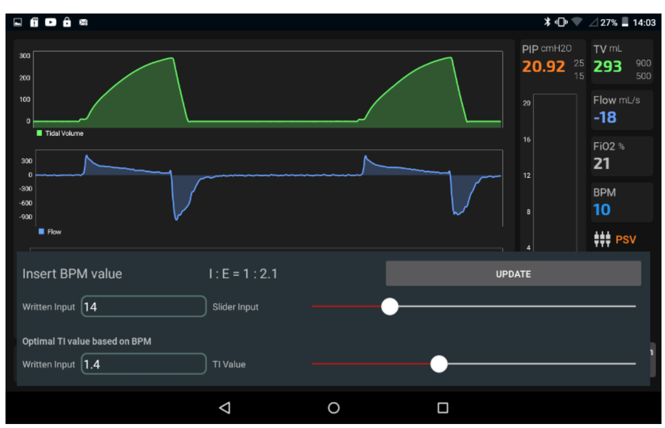

Figure 20: BPM can be set here

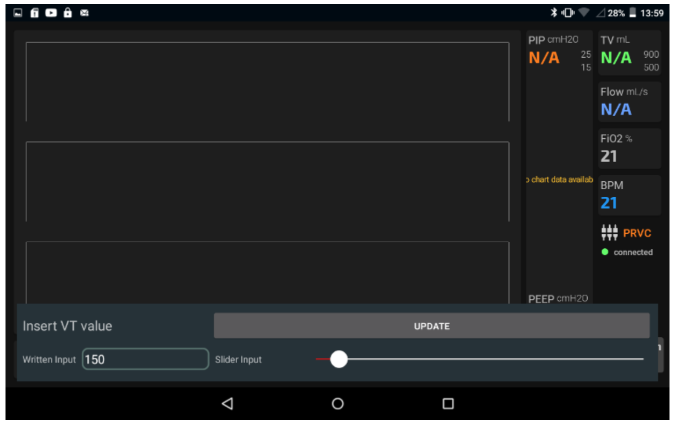

Figure 21: Tidal Volume can be set using the slider and updated

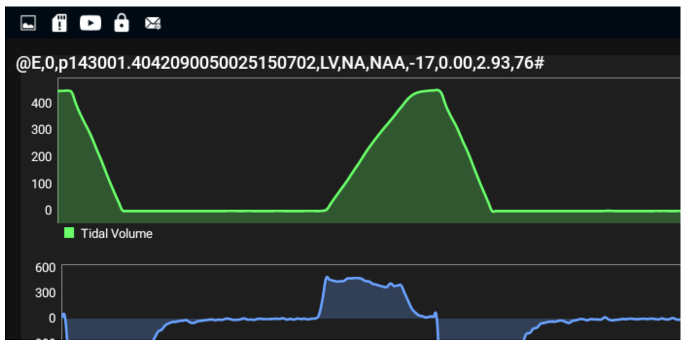

Figure 22: On the top left of the screen debugging string can be seen for developers

b) Firmware: plays a vital role proper integration between the software and hardware. It is highly crucial that the system runs uninterruptedly, maintaining the desired motor speed of the turbine and performing other necessary actuations while taking useful inputs from the different sensors and GUI simultaneously. The user inputs are sent from the Android application to the microcontroller via a Bluetooth module (HC-05) [10], which are then inputted into different control algorithms and other processes.

Control algorithms; Proportional Integral Derivative (PID) and feed-forward, have been designed and embedded into the firmware for the system to become robust to changes and provide effective outputs while operating the modes discussed before.

PID controller is a closed-loop control system with feedback control loop that evaluates the error based on the setpoint provided to the system [Figure 24]. To minimize the error, each part of the controller of P, I and D works hand in hand and then output is generated with each iteration while sending a feedback signal back to the system. [Figure 23] represents a control equation for PID, where the error [e(t)] is manipulated through the three constants; K_p, K_i and K_d. Not all of the PID is used; for instance, we have just used a ‘P’ controller in our ventilator.

$$motorSpeed = motorSpeed + volumeError * Kp \quad \text{----(2)}$$

The above [Equation 2] is implemented in order to precisely regulate the motor speed, the controller adjusts the value of the ‘volumeError’ in this formula. The value of Kp is determined after numerous trials and errors.

Only one of the representations of PID might be sufficient.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

Figure 23: Control equation of PID [11]

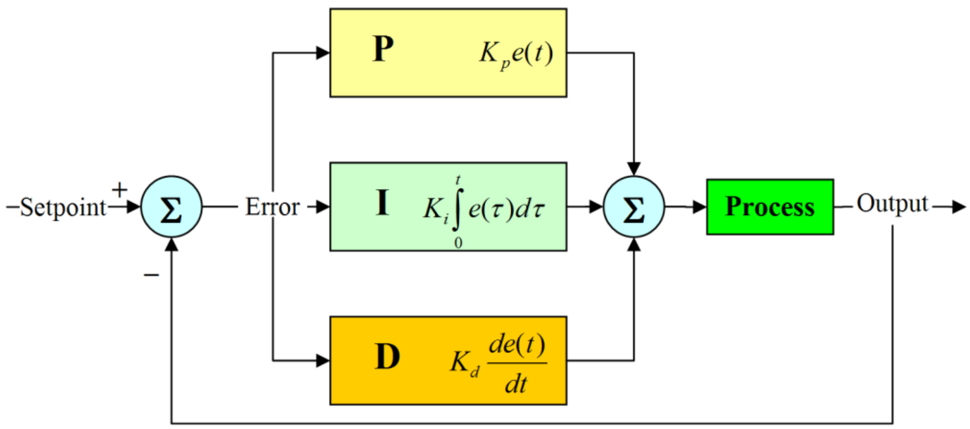

Figure 24: PID controller architecture [12]

While many of the controllers are designed so that the error generated due to disturbances and other factors can be reduced, feedforward control provides a direct solution to the control problem compensating the controlled variable before it even deviates from the set point by measuring the disturbances. The input to this control is determined in our system through rigorous testing in simulating various situations; value of ‘motorSpeed’ is the input given into the system using [Equation 2].

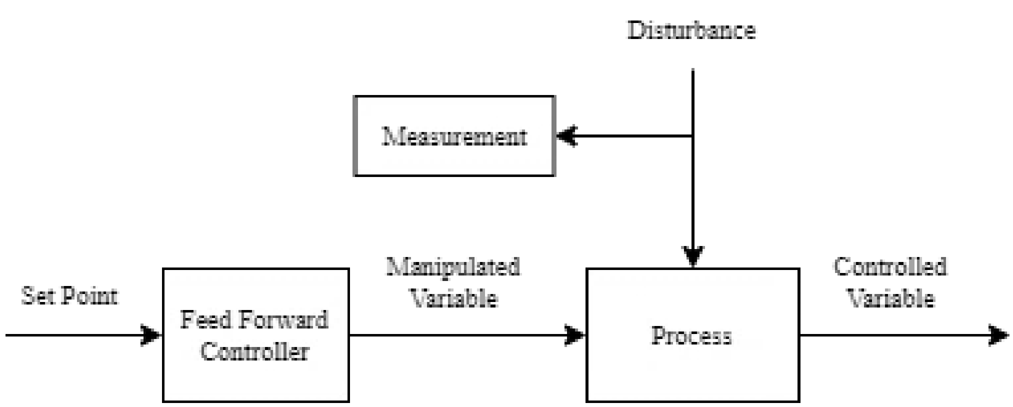

Figure 25: Feedforward Control Algorithm

Firmware is solely responsible for conducting all the calculations, switching between different modes while maintaining all the safety procedures and finally triggering necessary alarms if necessary.

While processing calculations on embedded systems we need to keep in mind a lot of things, such as, the memory occupied to run each of the functions, not to run out of the heap memory, etc. in order to run programs efficiently. Breathe cycle time is a very important factor for the system to proceed to the next steps is calculated by [Equation 3] and it is a summation of both inspiration and expiration times [Equation 4].

$$Breathe\ Cycle\ Time, BCT = \frac{60}{BPM} \quad \text{--(3)}$$

$$Breathe\ = Insp\ time + Expi\ time \quad \text{--(4)}$$

A predetermined data string frame is used for communication between the firmware and the Android tablet, for example, "v061502.0042090050025150702". Such a string is then broken-down using substring [22] to retrieve important data and put it into respective variables for the system to process later on.

i. Sensor data filtration methodology

   Data is being retrieved continuously from the system but however state-of-the-art hardware is, there will always be some noise that calls for some filtration before being used into any algorithm. The filtration algorithm is designed in the following steps:

   a. Data array- Sensor data (x) is accumulated into an array sequentially according to its array size.
   b. Mean ($\bar{x}$) is then calculated from the summation of the array values and data size using [Equation 5].

   $$\bar{x} = \frac{\sum x}{n} \quad \text{--(5)}$$

   c. Standard deviation 'stdDev' in [Equation 6] helps us to find out the extent of spreading of the sensor values and it is calculated using this formula [Equation 5].

   $$stdDev = \sqrt{\frac{\Sigma|x-\bar{x}|^2}{n}} \quad \text{--(6)}$$

   d. The standard deviation is then used to draw a virtual boundary/ envelope around the mean data; giving a better sense of the data to be filtered. We will be calling this envelope 'stdShrink', which is calculated using [Equation 7]. The shrinkPercent can be tweaked to get the desired stdShrink.

   $$stdShrink\ =\ stdDev\ -\ (stdDev\ *\ shrinkPercent) \quad \text{--(7)}$$

   [Figure 25] shows better representation of the data that needs to be filtered. Existing sensor data are then checked if they are within the 'DataBand'; $\bar{x} \pm stdShrink$. Data falling within this band would be summed to a new variable called filterAdd.

e. Finally, another mean is again calculated from the filtered data and then this mean shall be used by other modules of the system. These steps enable us to not only filter unwanted data spikes and noises created by the sensor, hence, assuring value with higher accuracy.

ii. Modes of Ventilation:
To provide seamless change in modes, the android app is used as a primary GUI where the mode is selected by the doctors.

a. PRVC- In this mode, firstly the targeted volume, breaths per minute (BPM), inspiration time (T_I), PEEP value and alarm settings are taken into the system via the data string.
Upon entering the inhale state, the motor starts with a speed; feed-forward value from the feed-forward algorithm, that has been hardcoded into the embedded system. The motor speed is then adjusted every cycle by running the PID algorithm with feed-forward algorithm in addition. Afterwards, peak inspiratory pressure (PIP) and other alarm parameters are checked and respective alarms are triggered if necessary.
A value for PEEP is taken in from doctors. During the exhale state, the motor starts with a feed-forwarded motor speed to maintain the specific PEEP value PID is again put into work. Alarm conditions are checked again and triggered accordingly.
Finally, if the conditions for the SIMV mode are met, inspiration mode is started accepting the spontaneous breath of the patient.

b. PCV- During this mode, the motor again starts with an initial speed that has been feed-forwarded beforehand and here this value dynamically changes in every loop. Within the inhale state, PID is used to maintain the motor speed [Equation 2] to keep the desired pressure by comparing raw pressure sensor data. The PIP is determined using the 'Sensor Data Filtration Methodology' [2.4.b.i]. Algorithm is then recalibrated in accordance with the newly found value from the filtered sensor dataset. New base value (feed forward) for the motor is calibrated and the motor speed is maintained accordingly using PID. Other alarm parameters and patient circuit disconnection are checked and are triggered if necessary.
During exhale state, the motor starts with a feed-forwarded motor speed to maintain the specific PEEP value set by the doctor. The motor speed needed is maintained using a PID algorithm. Subsequently Alarm conditions are checked again and triggered accordingly.
If the conditions for the SIMV mode is met, inspiration mode is started.

c. SIMV- is more like a feature of our system that can be turned on or off. With the inputs of T_i and respiratory rate from the doctor, the breath cycle is calculated using [Equation 3]. Then the 'nonSyncTime' is calculated from the breath cycle using [Equation 8]. We have considered the sync time to be after 90% of the breath cycle time but this can easily be changed in the code that we have provided. In order to calculate the time window for the SIMV, 'syncTime' is calculated using the [Equation 9].

$$nonSyncTime = breathCycleTime * 0.9 \text{--(8)}$$

$$SyncTime = nonSyncTime - Ti \text{ --(9)}$$

If a patient tries to breathe within this 'syncTime' [Figure 26], flow rate is checked and if it is greater than 20 ml, breath is given. Value for the flow rate for the trigger is taken from the doctor. The sensitivity of the trigger can be tuned by adjusting the value of the flow rate by the. Same type of mechanism can be implemented using a Pressure sensor. We have however implemented only an expiratory time synchronization window with flow rate inspection.

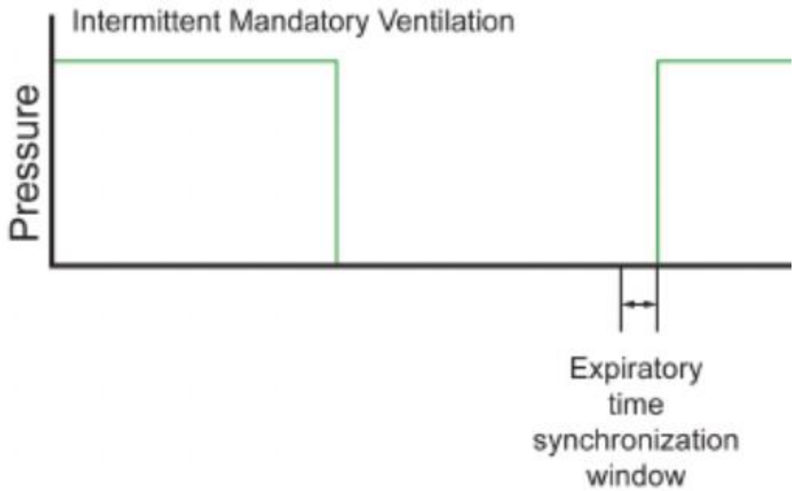

Figure 26: SIMV synchronization window [23]

d. Patient Circuit Disconnection Algorithm: This is the feature that holds quite a bit of significance. During the inhale state if PIP is found to be less than 5 cm H_2O, a disconnection is triggered. Here, the flow increases as the pressure cannot be built in an open patient-circuit. Both audio and visual alarms are given out in the form of a buzzer from an android tablet and distinguishable text on the GUI.

As a safety feature we decrease the motor speed to the minimum level so that when the patient circuit is reconnected, it can be detected and the pressure build-up becomes easier and faster.

Just to add a few more advantages of our device is that both the steps required for operation, start-up time are very less because the flow sensor does not need calibration and the fast calibration of the pressure sensor. However, air relay can be used for calibrating the pressure sensor after a certain interval to improve the performance further.

3. Design files

Design file name	File type	Open-source license
base 11	CAD files (STEP)	CC BY-SA 4.0
base 12	CAD files (STEP)	CC BY-SA 4.1
base 21	CAD files (STEP)	CC BY-SA 4.2
base 22	CAD files (STEP)	CC BY-SA 4.3
top 22	CAD files (STEP)	CC BY-SA 4.4
top 21	CAD files (STEP)	CC BY-SA 4.5
debug cover	CAD files (STEP)	CC BY-SA 4.6
charging port	CAD files (STEP)	CC BY-SA 4.7
top 11	CAD files (STEP)	CC BY-SA 4.8
top 12	CAD files (STEP)	CC BY-SA 4.9
front 11	CAD files (STEP)	CC BY-SA 4.10
back 11	CAD files (STEP)	CC BY-SA 4.11
back 12	CAD files (STEP)	CC BY-SA 4.12
front 12	CAD files (STEP)	CC BY-SA 4.13
Side Vent v1	CAD files (STEP)	CC BY-SA 4.14
Tab Cover 1	CAD files (STEP)	CC BY-SA 4.15
Tab Cover 2	CAD files (STEP)	CC BY-SA 4.16
Button1 v1	CAD files (STEP)	CC BY-SA 4.17
Button2 v1	CAD files (STEP)	CC BY-SA 4.18
Air Filter Vent v1	CAD files (STEP)	CC BY-SA 4.19
Front Extension 1 (Base)	CAD files (STEP)	CC BY-SA 4.20
Front Extension 1 (Attachment	CAD files (STEP)	CC BY-SA 4.21
Front Extension 2 v1	CAD files (STEP)	CC BY-SA 4.22
Pressure Hose Attachment v1	CAD files (STEP)	CC BY-SA 4.23
Indicator v1	CAD files (STEP)	CC BY-SA 4.24
Push Button v1	CAD files (STEP)	CC BY-SA 4.25
DMX 5pin v1	CAD files (STEP)	CC BY-SA 4.26
Rotary Encoder Module v1	CAD files (STEP)	CC BY-SA 4.27
Knob 20mm v1	CAD files (STEP)	CC BY-SA 4.28
Power Socket v1	CAD files (STEP)	CC BY-SA 4.29
Switch DPDT v1	CAD files (STEP)	CC BY-SA 4.30
DC Power Socket v1	CAD files (STEP)	CC BY-SA 4.31
LCD Holder 1	CAD files (STEP)	CC BY-SA 4.32

LCD Holder 2	CAD files (STEP)	CC BY-SA 4.33
LCD Holder 3	CAD files (STEP)	CC BY-SA 4.34
2.8_TFT_LCD_Display v1	CAD files (STEP)	CC BY-SA 4.35
3mm Pin	CAD files (STEP)	CC BY-SA 4.36
Tab v1	CAD files (STEP)	CC BY-SA 4.37
Holder1	CAD files (STEP)	CC BY-SA 4.38
Holder2	CAD files (STEP)	CC BY-SA 4.39
Support	CAD files (STEP)	CC BY-SA 4.40
Holder3	CAD files (STEP)	CC BY-SA 4.41
Outlet Body	CAD files (STEP)	CC BY-SA 4.42
Outlet Base	CAD files (STEP)	CC BY-SA 4.43
Exhaust Valve Gate v1	CAD files (STEP)	CC BY-SA 4.44
Stopper Bolt	CAD files (STEP)	CC BY-SA 4.45
Stopper Gasket	CAD files (STEP)	CC BY-SA 4.46
EOM(Base)	CAD files (STEP)	CC BY-SA 4.47
EOM (Push Body)	CAD files (STEP)	CC BY-SA 4.48
EOM (Servo Bracket)	CAD files (STEP)	CC BY-SA 4.49
Silicon Seal	CAD files (STEP)	CC BY-SA 4.50
Screw	CAD files (STEP)	CC BY-SA 4.51
EGV Base	CAD files (STEP)	CC BY-SA 4.52
MG995 v1	CAD files (STEP)	CC BY-SA 4.53
Servo Horn v1	CAD files (STEP)	CC BY-SA 4.54
Exhaust Gate Valve Coil Spring v1	CAD files (STEP)	CC BY-SA 4.55
One Way Valve Inlet	CAD files (STEP)	CC BY-SA 4.56
One Way Valve Outlet	CAD files (STEP)	CC BY-SA 4.57
Adapter_14mm (Long) v1	CAD files (STEP)	CC BY-SA 4.58
Adapter14mm v1	CAD files (STEP)	CC BY-SA 4.59
WM7040 Pump v1	CAD files (STEP)	CC BY-SA 4.60
HME Body	CAD files (STEP)	CC BY-SA 4.61
HME Cover	CAD files (STEP)	CC BY-SA 4.62
HME Inlet Mesh	CAD files (STEP)	CC BY-SA 4.63
Inlet Body v1	CAD files (STEP)	CC BY-SA 4.64
Proportional Valve Body	CAD files (STEP)	CC BY-SA 4.65
Proportional Valve Cover	CAD files (STEP)	CC BY-SA 4.66
Adapter_14mm (Short) v1	CAD files (STEP)	CC BY-SA 4.67
Adapter_14mm (Pump Outlet) v1	CAD files (STEP)	CC BY-SA 4.68

Micro Tower 9g v1	CAD files (STEP)	CC BY-SA 4.69
peep inlet	CAD files (STEP)	CC BY-SA 4.70
peep outlet	CAD files (STEP)	CC BY-SA 4.71
Peep Extension	CAD files (STEP)	CC BY-SA 4.72
Peep Valve Attachment	CAD files (STEP)	CC BY-SA 4.73
Hose_OXY	CAD files (STEP)	CC BY-SA 4.74
Hose_Release	CAD files (STEP)	CC BY-SA 4.75
Peep Hose	CAD files (STEP)	CC BY-SA 4.76
OXY Nozzle Body	CAD files (STEP)	CC BY-SA 4.77
OXY Nozzle Holder	CAD files (STEP)	CC BY-SA 4.78
PSU mount v1	CAD files (STEP)	CC BY-SA 4.79
SMPS v1	CAD files (STEP)	CC BY-SA 4.80
3S Battery Holder v1	CAD files (STEP)	CC BY-SA 4.81
PANASONIC_NCR-18650B v1	CAD files (STEP)	CC BY-SA 4.82
Holder	CAD files (STEP)	CC BY-SA 4.83
PSU Holder Tab v1	CAD files (STEP)	CC BY-SA 4.84

4. Bill of Materials

Designator	Component	Number	Cost per unit (USD)	Total cost (USD)	Source of materials
WM7040 Pump v1	Turbine	1	53	53	https://www.aliexpress.com/item/10000048326249.html?spm=a2g0s.8937460.0.0.438c2e0ePyDUwe
	SFM3300-250-D Flow Sensor	1	45	45	https://www.digikey.com/en/products/detail/sensirion-ag/SFM3300-250-D/9857673
	Pressure Sensor-MPX2010DP	1	19	19	https://www.digikey.com/en/products/detail/nxp-usa-inc/MPX2010DP/410890?s=N4IgTCBcDaILYAcAeYAMBGVATBIC6AvkA
	Arduino Mega	1	9.49	9.49	https://cutt.ly/zmbwZLd
	Buck Converter - XL4015	3	1.3	3.9	https://cutt.ly/xl4015
	Buck Converter - LM2596	1	1	1	https://cutt.ly/lm2596
	Digital Encoder	1	1	1	https://cutt.ly/1mv4wQJ
MG995 v1	MG995 Servo	1		2	https://cutt.ly/Smv7uFC
Micro Tower 9g v1	Sg90 Micro Servo	1	1	1	https://cutt.ly/bmv7vfv
	Op Amp LM358	1	0.53	0.53	https://www.digikey.com/en/products/detail/texas-instruments/LM358PE4/1510269

	Diode MBR20100	1	1.07	1.07	https://www.digikey.com/en/products/detail/smc-diode-solutions/MBR20100/6022125
	4.7K Ohms Resistor	2	0.26	0.52	https://www.digikey.com/en/products/detail/vishay-beyschlag-draloric-bc-components/MBB02070C4701FCT00/7351222
	10K Ohm Metal Film Resistor	2	0.4	0.8	https://www.digikey.com/en/products/detail/vishay-beyschlag-draloric-bc-components/PR02000201002JA100/7351718
	1M Ohm Resistor	1	0.26	0.26	https://www.digikey.com/en/products/detail/vishay-beyschlag-draloric-bc-components/MRS25000C1004FRP00/5063981
	Capacitor 104/100nF	2	0.45	0.9	https://www.digikey.com/en/products/detail/vishay-beyschlag-draloric-bc-components/K104K15X7RF53H5G/13279844
Tab v1	Android Tab 10.11" Touch Screen	1	95	95	https://www.bdstall.com/details/5-star-2gb-ram-tablet-pc-50422/
2.8_TFT_LCD_Display v1	3.2 TFT Display	1	11.61	11.61	https://cutt.ly/omv3c9n
PANASONIC_NCR-18650B v1	18650 Battery	6	3.65	21.9	https://cutt.ly/xmv9Gkt
	BMS 3S1P	2	1	2	https://cutt.ly/Omv8Q3u

3S Battery Holder v1	Battery Spacer	4	0.24	0.96	https://cutt.ly/dm6nxyz
	USB Cable 1 meter	1	3	3	https://www.aliexpress.com/item/4001177937131.html?spm=a2g0o.productlist.0.0.4b222681PWZUVC&aem_p4p_detail=202107041642053645446303884900061129794
DMX 5pin v1	GX12-5 Connector	1	6	6	https://www.banggood.com/Electronic-Soldering-Iron-GX12-5-Connector-T12-Aviation-Head-Mini-Aviation-Male-DIY-Soldering-Kits-p-1153612.html?cur_warehouse=CN&rmmds=buy
	Pogo Pins	1	4.76	4.76	https://www.digikey.com/en/products/detail/mill-max-manufacturing-corp/816-22-006-10-001101/6149708
	Solderless docking type 4P, 5.08MM screw terminal	3	0.5	1.5	https://cutt.ly/4mx8ghB
DC Power Socket v1	DC Jack Female	1	0.2	0.2	https://cutt.ly/rmv8rFg
	USB Type A Standard Port Female	1	0.1	0.1	https://cutt.ly/Fmx8UJH
	KF301-5.0-2P Green Connector	1	0.1	0.1	https://cutt.ly/Tmx4yFU

	Straight Male Pin Header	1	0.1	0.1	https://cutt.ly/Kmx7BCz
	Male Header L Shape	1	0.8	0.8	https://cutt.ly/rmv7ufC
	L Connector				
Power Socket v1	Power Socket	1	0.3	0.3	https://cutt.ly/8mv8Ehn
	TO220 Heat Sink	1	0.5	0.5	https://cutt.ly/qmx7H1O
	Heat Shrink	1	0.32	0.32	https://cutt.ly/Cmv4PAG
	Ventilator Patient Circuit	1	6	6	https://bmabazar.com/shop/ventilator-patient-circuit/
	Viral Filter	1	5	5	https://bmabazar.com/shop/viral-filter/
	PLA Filament for 3D Printing	3	23	69	https://www.sunlu.com/products/new-pla-plus-pla-1-75-filament-1kg-2-2lbs?variant=31987368788054
Screw	Screw Assortment Kit	1	20	20	https://www.aliexpress.com/item/33033655207.html?spm=a2g0o.productlist.0.0.6e2077dcdv92xc&aem_p4p_detail=20210705004947357246090160200 57207816
	Epoxy Glue	5	0.91	4.55	https://www.aliexpress.com/item/1005002511958246.html?spm=a2g0o.productlist.0.0.405f1e3cUZ4UkH&algo_pvid=85d47592-8cf0-4f68-99ac-b1c1b101c348&algo_exp_id=85d47592-8cf0-4f68-99ac-b1c1b101c348-3

Exhaust Gate Valve Coil Spring v1	Coil Spring:	1	2.29	2.29	https://www.aliexpress.com/item/4000411749473.html?spm=a2g0o.productlist.0.0.e4573839rqzi4A&algo_pvid=471f3601-583e-449f-83e7-bbaa1fb64c96&algo_exp_id=471f3601-583e-449f-83e7-bbaa1fb64c96-1
	(Length 25mm, Outer diameter 5mm, Wire thickness 0.3mm)				
	10 pcs Pack				
Stopper Gasket	Rubber O-ring kit	1	4.35	4.35	https://cutt.ly/nmWW9V6
	3mm Linear Rod (2 Pcs kit, 3mm*100mm)	1	1.2	1.2	https://cutt.ly/amSgoMs
Hose_OXY, Hose_Release	Silicon Tube 1 foot	2	1.2	2.4	https://cutt.ly/dm6mwlX
Silicon Seal	Silicone Sheet	1	8.95	8.95	https://cutt.ly/0mbtd5E
Peep Hose	4mm Medical grade pipe	1	4.99	4.99	https://www.aliexpress.com/item/4000415201699.html?spm=a2g0o.productlist.0.0.1f6b709bzdHdIe&algo_pvid=fde0c927-c1c6-4d83-9afe-dd2b004d985d&algo_exp_id=fde0c927-c1c6-4d83-9afe-dd2b004d985d-1
	Heat Shrink Tube	1	2.2	2.2	https://cutt.ly/7mv5G4w

Peep Valve Attachment , Front Extension 1 (Attachment)	Extension Tube (4mm)	1	8.86	8.86	https://cutt.ly/emSfAHN

5. Build Instructions

5.1 Casing

1. Glue the base parts (base 11, 12 21 and 22).

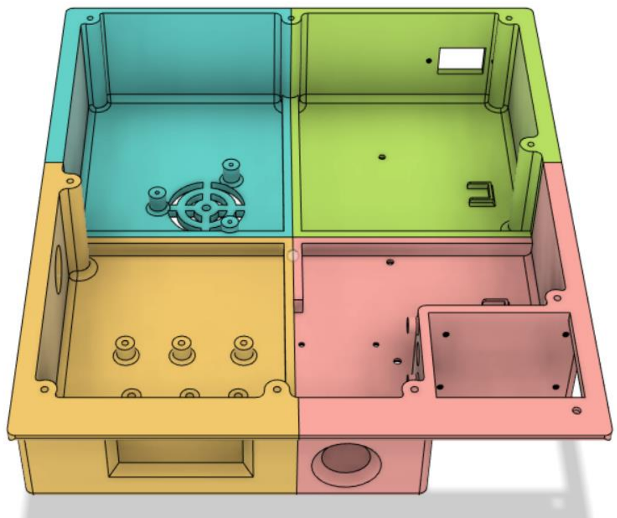

Figure 26: Base Parts

2. Glue down the ‘PSU Holder Tabs’ with the Casing Base (Base 11 and 22)

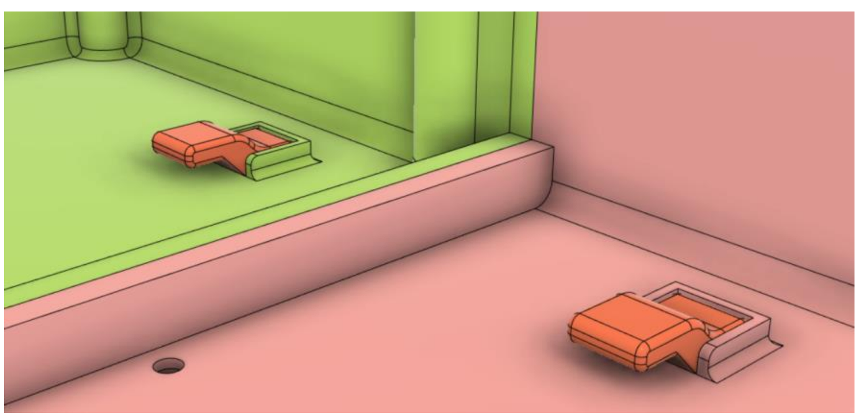

Figure 27: PSU Holder Tabs

3. Glue the Front Extension 1 (Base) and Front Extension 1 (Attachment) together

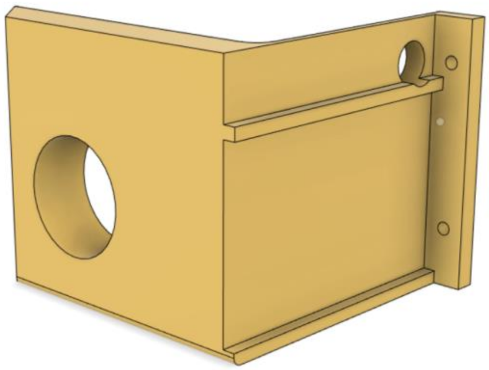

Figure 28: Front Extension 1 (Base)

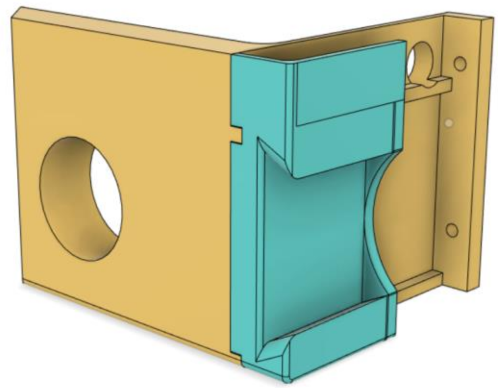

Figure 29: Front Extension 1 (Attachment)

4. Glue the Front Extension 1 as a whole to the Base 11

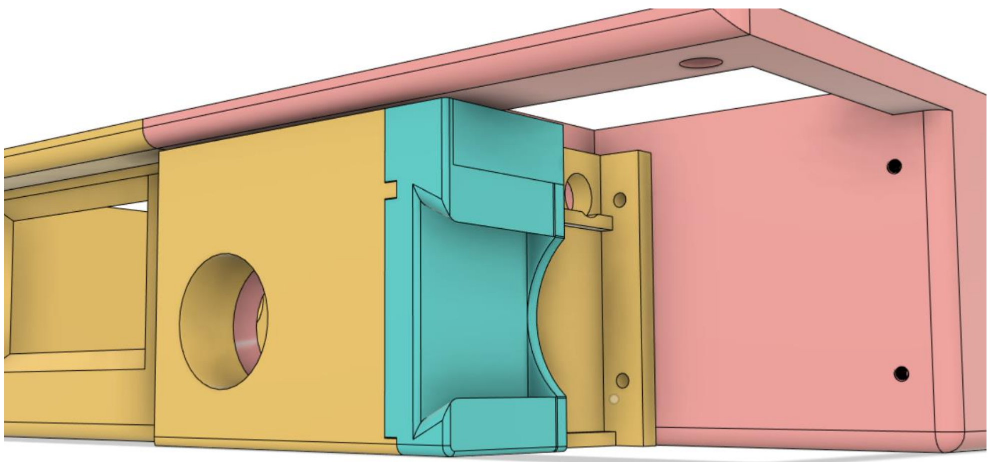

Figure 30: Front Extension 1 attached with the Base 11

5. Attach the 'Pressure Hose Attachment' with the 'Front Extension 2' and then glue the entire assembly with the 'Base 11'

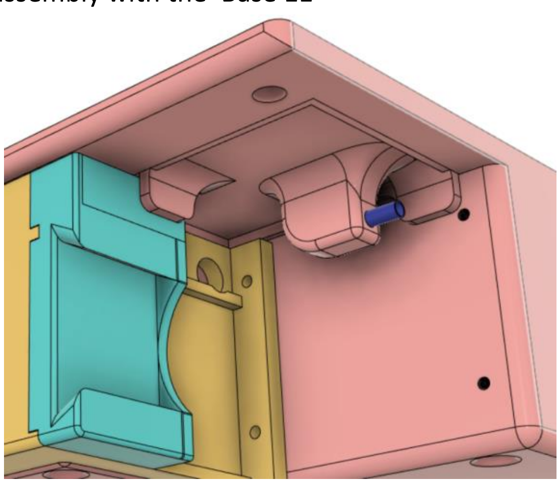

Figure 31: Bottom view of Front Extension 2

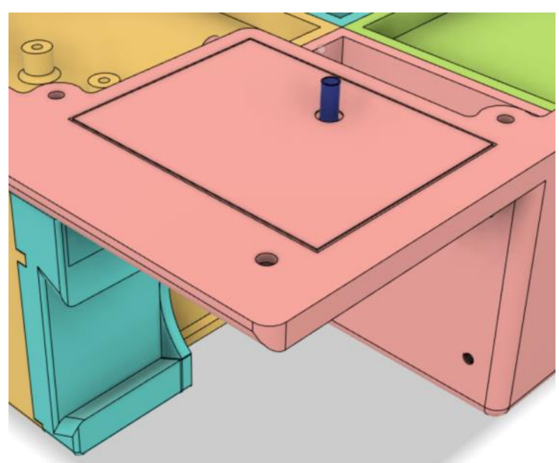

Figure 32: Top view of Front Extension 2

6. Glue the ‘Air Filter Vent’ with the ‘Base 12’

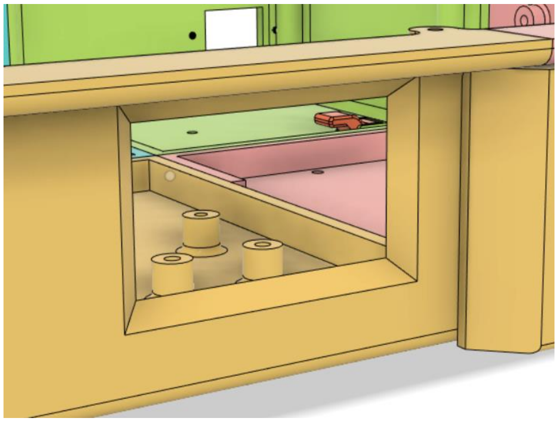

Figure 33: Without vent

Figure 34: With vent

7. Glue the Top Parts (Top 11, 12, 21 and 22)

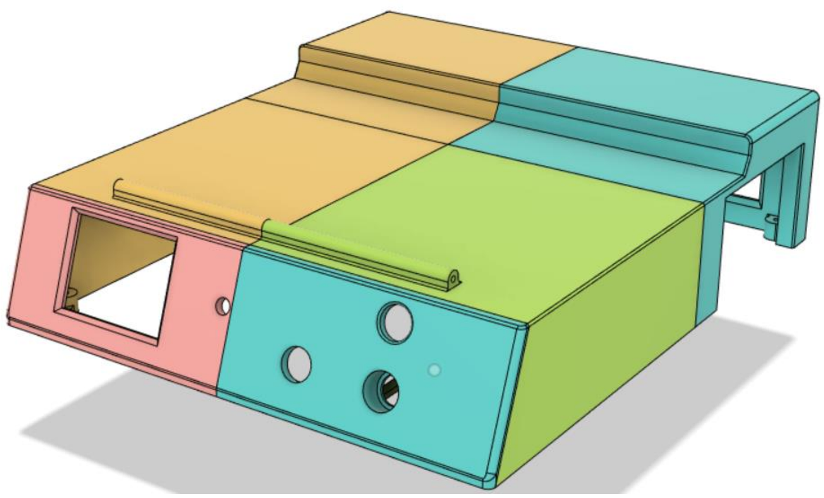

Figure 35: Top Parts glued together

8. Glue the Side Vents to the Top Parts (Top 21 and 22)

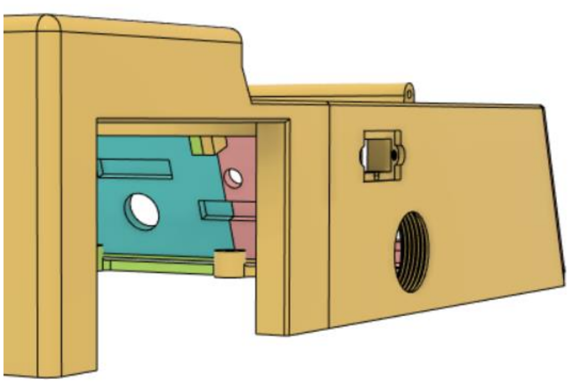

Figure 36: Without side vent

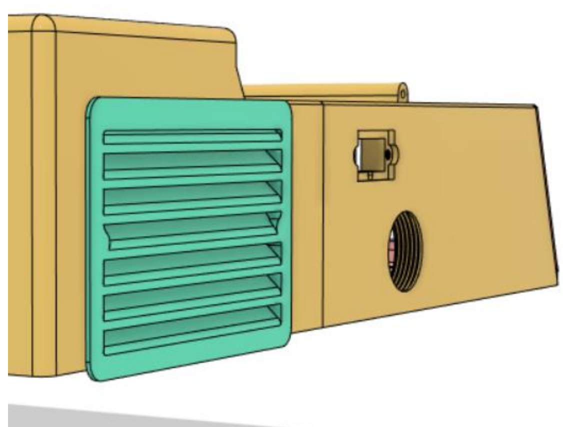

Figure 37: With side vent

9. Glue the LCD Holder Parts (1 and 3) with the Top 12

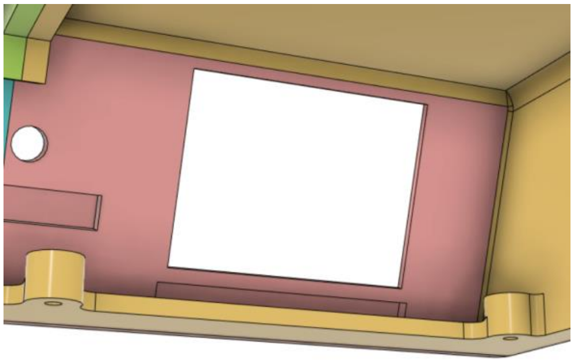

Figure 38: Without LCD Holder

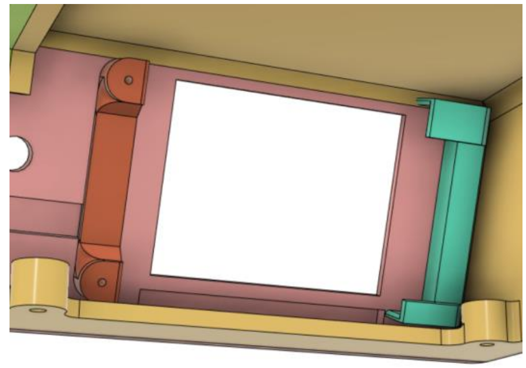

Figure 39: With LCD Holder

10. Glue the Tab Covers: Front 11 (a and b) and Front 12 (a and b)

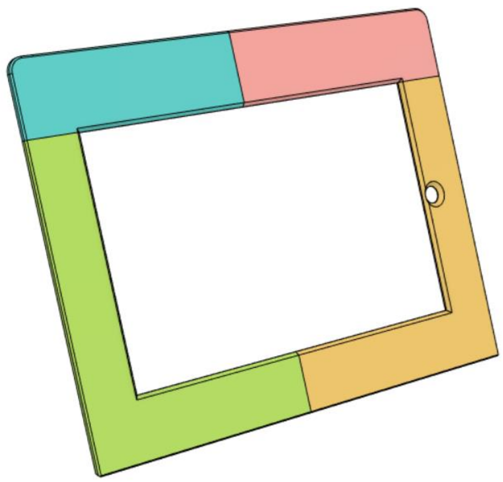

Figure 40: Front 11 and 12 are glued together

11. Glue the Tab Covers: Back 11 and 12

Figure 41: Back 11 and 12

12. Glue the Tab Covers: Front and Back parts

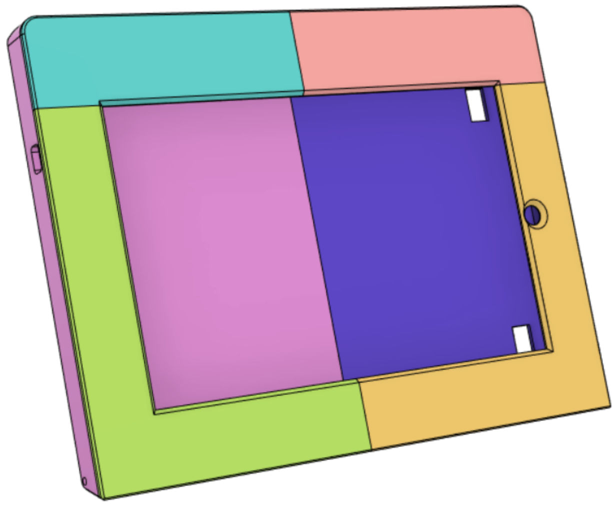

Figure 42: Front and Back Tab Covers

13. Glue the Tab Cover Top parts (1 and 2)

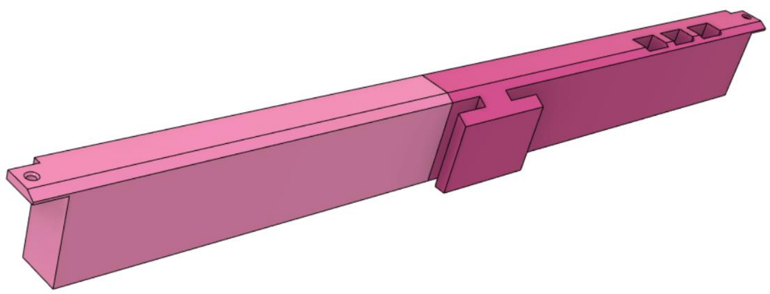

Figure 43: Tab Cover Top parts (1 and 2)

14. Let the glue to dry and set the pieces together
15. Install the Power Socket with Base 22, Switch DPDT and DC Power Socket with Top 21 and Debug Cover with Top 22

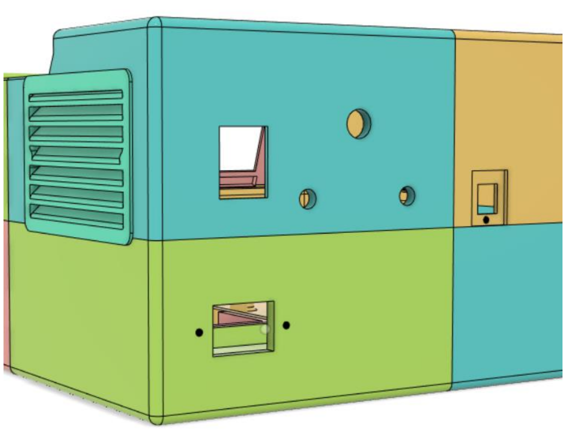

Figure 44: Rear view without ports and sockets

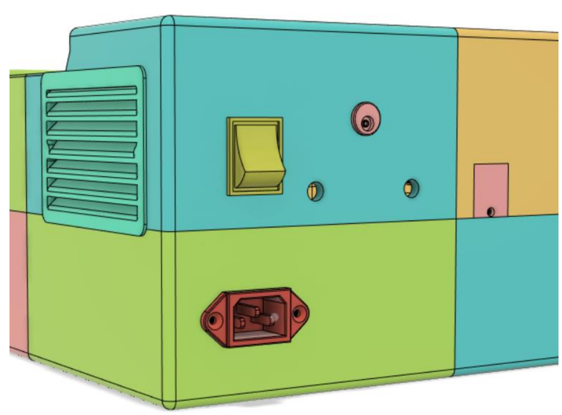

Figure 45: Rear view with installments

16. Slide in the ‘2.8 TFT LCD Display’ into the LCD Holder 1 and then place LCD Holder 2 on the top left of the display and screw it into place

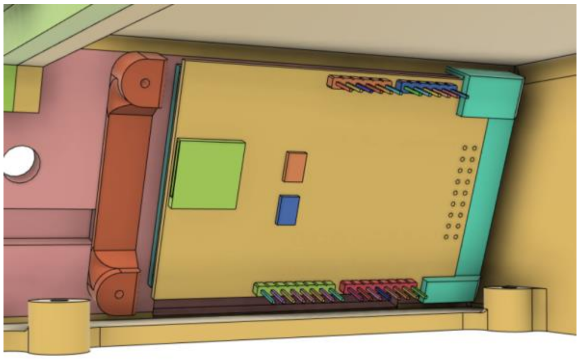

Figure 46: LCD display slid into place

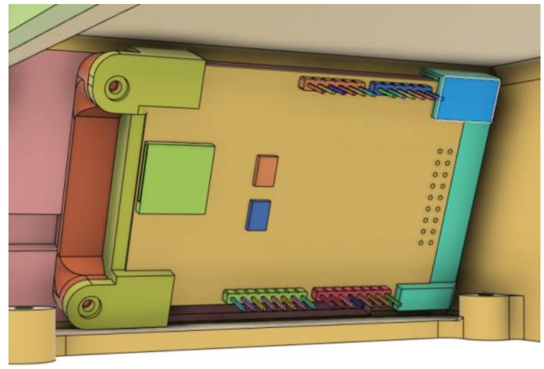

Figure 47: LCD Holder 2 on the top left of the display, ready to be screwed into place

17. Install the ‘Rotary Encoder’, ‘Knob 20 mm’, ‘Indicator’, ‘Push Button’ and ‘DMX 5 pin socket’ with the Top 11 and 22

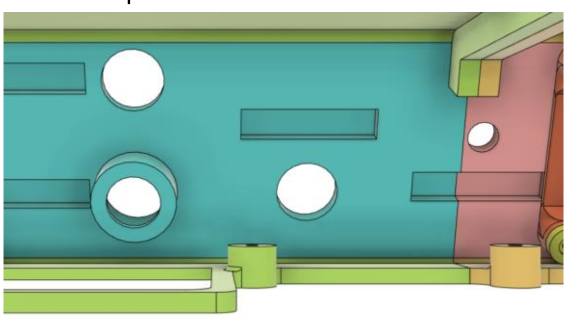

Figure 48: Without parts

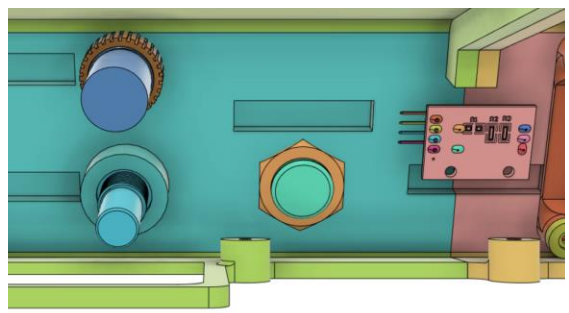

Figure 49: Installed parts (rear view)

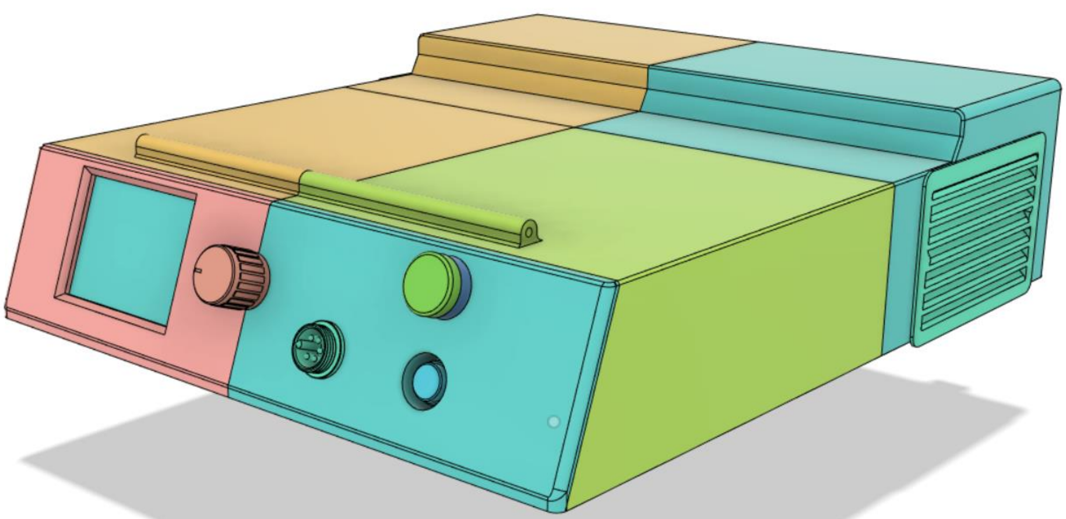

Figure 50: Installed parts (Front view)

18. Two 3 mm Linear Rods have been inserted into the Hinge part of Back11 and Back12 from the opposite sides for the Tab Cover to rotate about the linear rod's axis

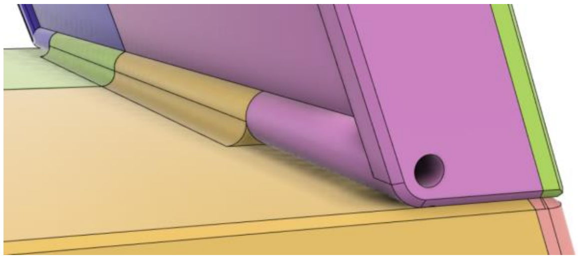

Figure 51: Without 3mm linear rods

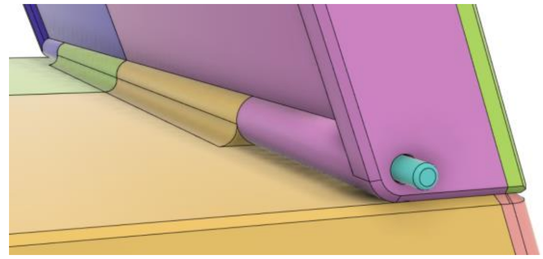

Figure 52: With 3mm linear rods

19. Insert the Tab into the Tab Cover Front and Back and place the buttons in the specific places and then screw down the Tab Cover Top

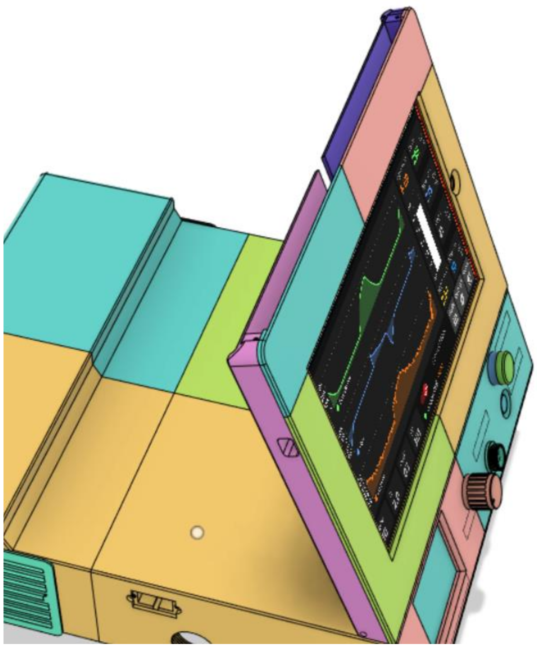

Figure 53: Tablet inserted into place

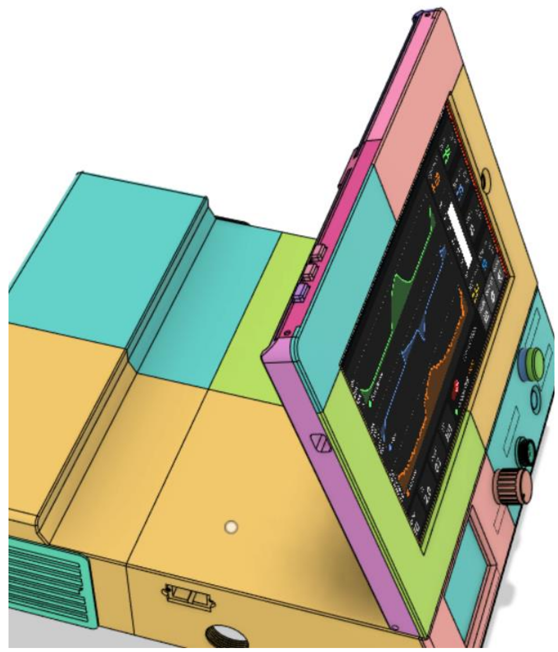

Figure 54: With Tab Cover Top attached to holder tablet into place

20. Glue the Holder 2 with Top 11 and Top 21, Holder 1 with Back 11 and Holder 3 with Top 12 and Top 22. Support 1 is used to hold the tablet at an angle for better view of the display. Distance between the Holder 1 and 2 and distance between 2 and 3 are kept the same so that the tablet support can be kept with the case when not in use.

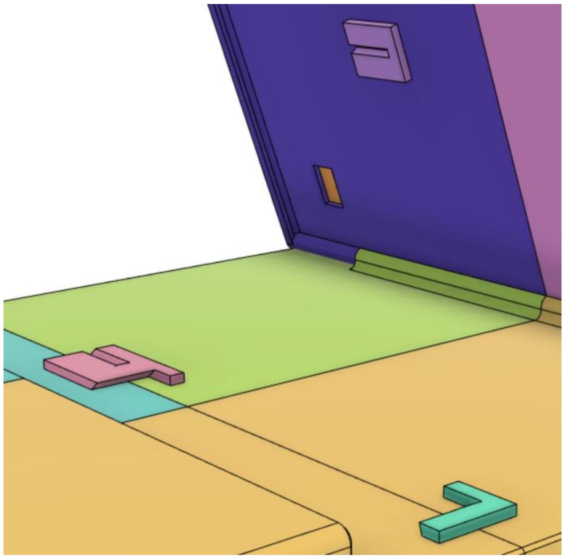

Figure 55: Holder 2 with Top 11 and Top 21, Holder 1 with Back 11 and Holder 3 with Top 12 and Top 22

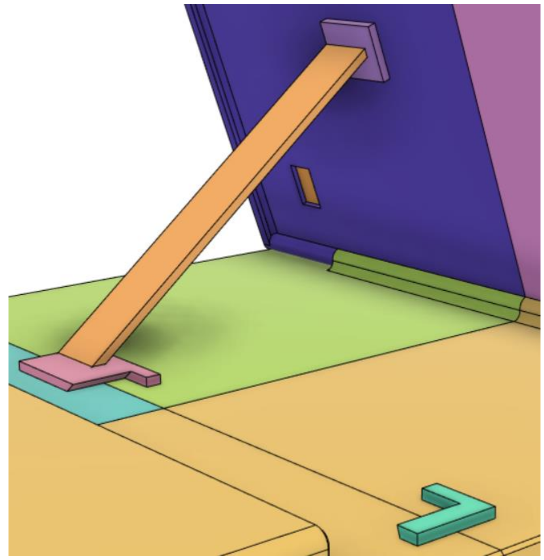

Figure 56: Support 1 placed in between the Holder 1 and 2

21. Top Full Assembly

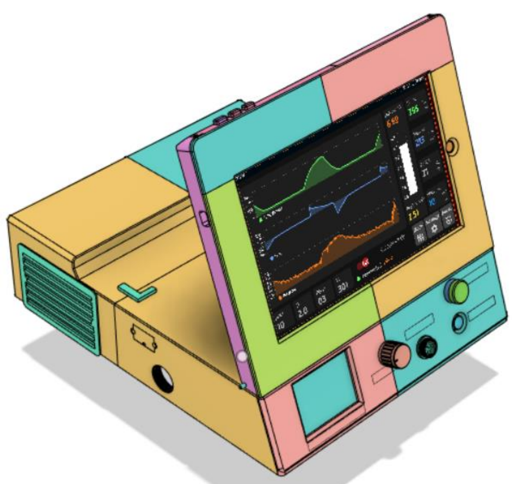

Figure 57: Top Assembly (Front View)

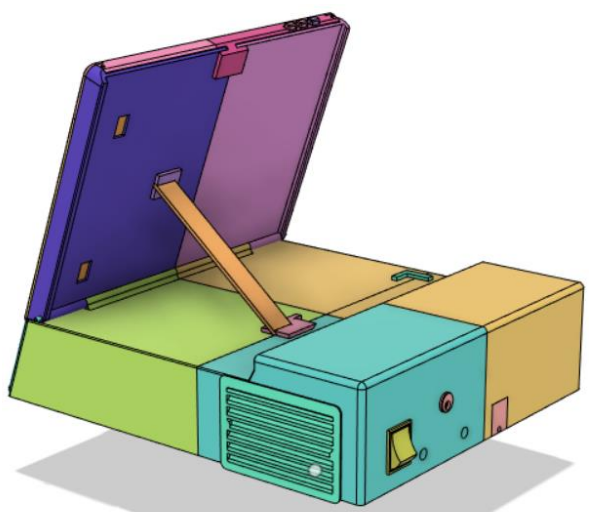

Figure 58: Top Assembly (Rear View)

2.2 One-way Valve

1. Glue the One-way valve inlet and outlet

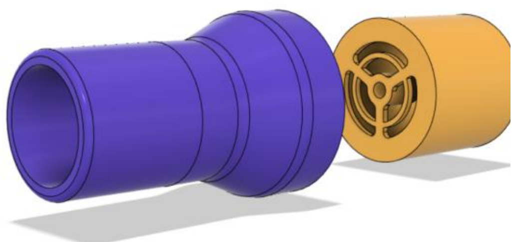

Figure 59: Parts before gluing them together

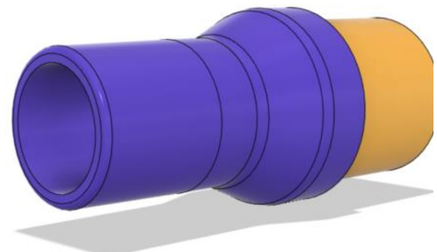

Figure 60: Parts after gluing them together

2. Insert the Silicone Diaphragm through the Valve outlet side and place on the mesh and screw it down

Figure 61: Before screwing in the diaphragm

Figure 62: After screwing in the diaphragm

2.3 PEEP valve

1. Glue the outlet and the attachment

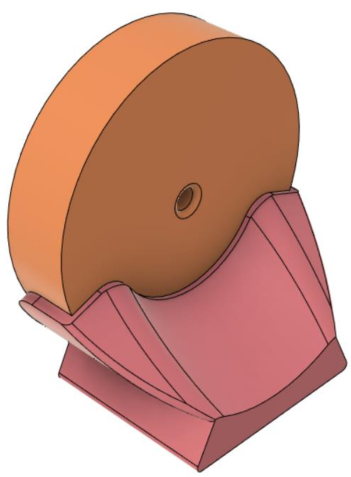

Figure 63: Gluing the outlet and attachment

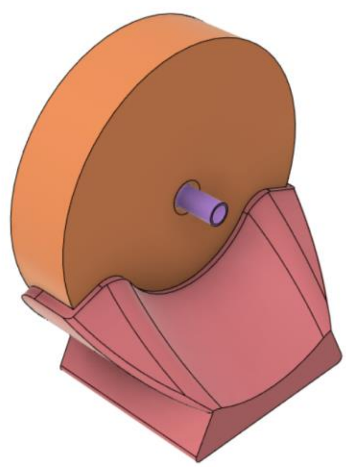

Figure 64: Attaching the Peep Extension with the outlet

2. Place the Diaphragm on the Inlet and screw it

Figure 65: Before screwing the diaphragm

Figure 66: After screwing the diaphragm

3. After the glue dries out, screw the Inlet and Outlet

Figure 67: Before screwing the pieces together

Figure 68: After screwing the parts

2.4 Pressure Release Mechanism (PRM)

1. Glue the EOM (Servo Bracket) and Base

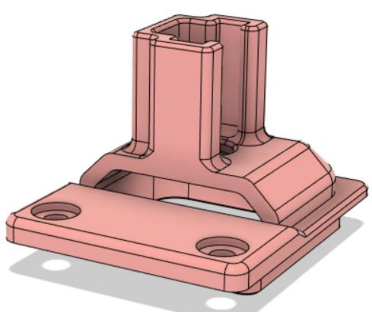

Figure 69: EOM (Base)

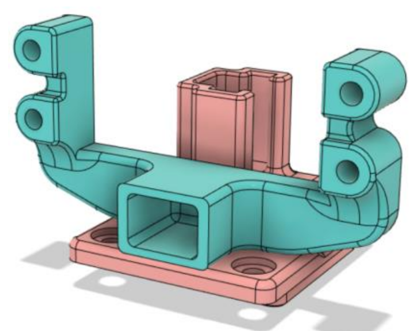

Figure 70: EOM (Servo Bracket) glued to EOM (Base)

2. Glue the exhaust valve gate and Outlet Body

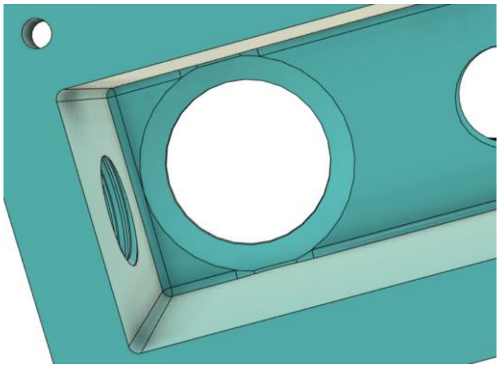

Figure 71: Outlet Body

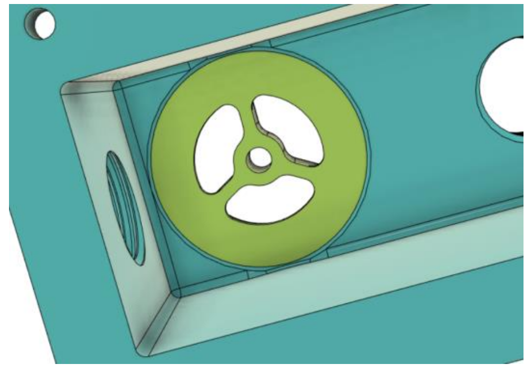

Figure 72: Exhaust Valve Gate glued to the Outlet Body

3. Glue the Screw and EGV Base and then glue the Silicone Seal with the EGV Base

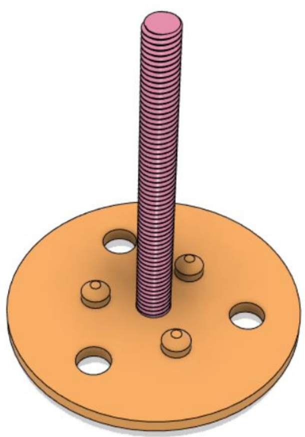

Figure 73: Screw and EGV Base glued together

Figure 74: Silicone attached with the EGV Base

4. Place the Stopper Gasket with the Stopper Bolt and screw it with the Outlet Body

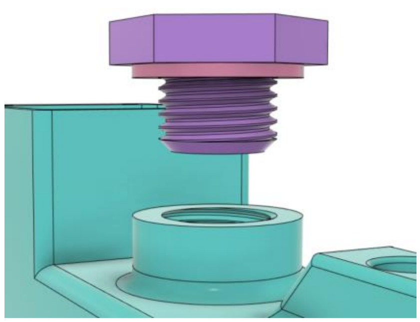

Figure 75: Place the Stopper Gasket on the Stopper Bolt

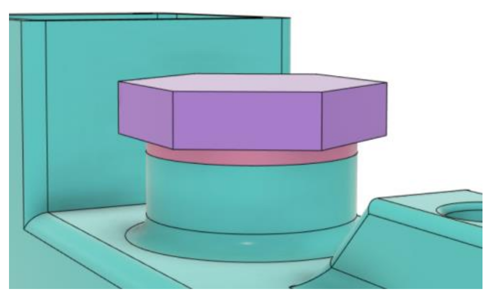

Figure 76: Screw the Stopper with the Outlet Body

5. Screw in the 14 mm Adapters to the Outlet Body

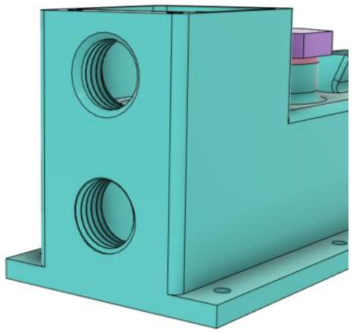

Figure 77: Outlet Body without the adapters

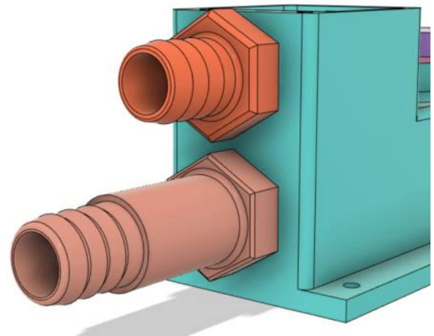

Figure 78: Outlet Body with the adapters

6. Put the Exhaust Valve through the Exhaust Valve gate from the bottom of the Outlet Body and the Exhaust Gate Valve Coil Spring through the protruded screw from the top of the Outlet Body

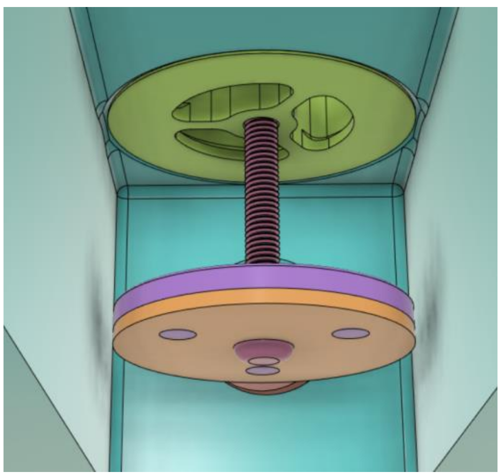

Figure 79: Exhaust Valve through the Exhaust Valve gate

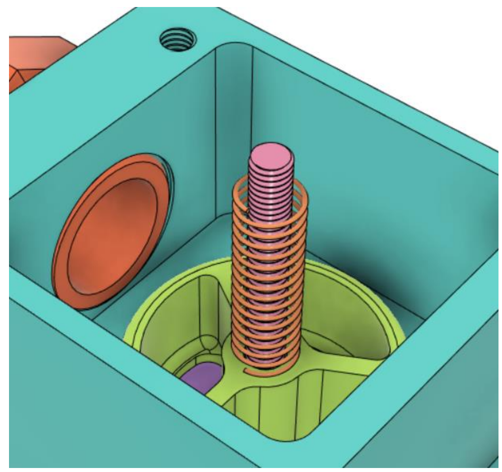

Figure 80: Exhaust Gate Valve Coil Spring

7. Put the Exhaust Opening Mechanism (EOM Base and Servo bracket) on the top of the Outlet Body and then EOM (Push Body) to be screwed through the Exhaust Valve.

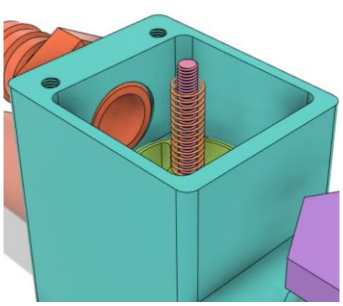

Figure 81: Outlet Body before EOM (Base and Servo Bracket) attached

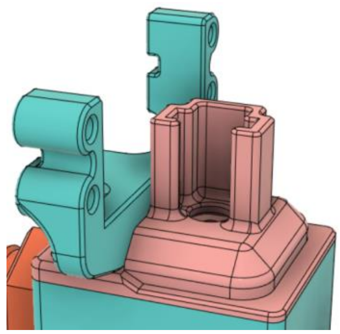

Figure 82: Outlet Body after EOM (Base and Servo Bracket) attached

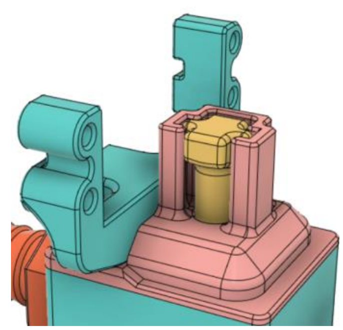

Figure 83: Outlet Body after EOM (Push Body) attached

8. Screw the Servo with the EOM (Servo Bracket) and install the Servo horn with the Servo

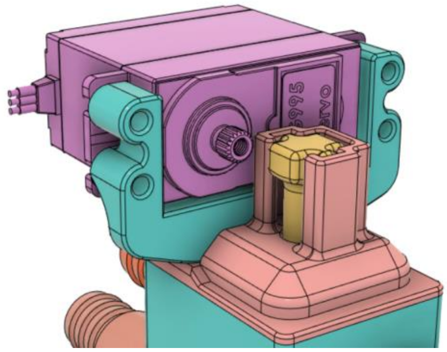

Figure 84: Servo installed to the EOM (Servo Bracket)

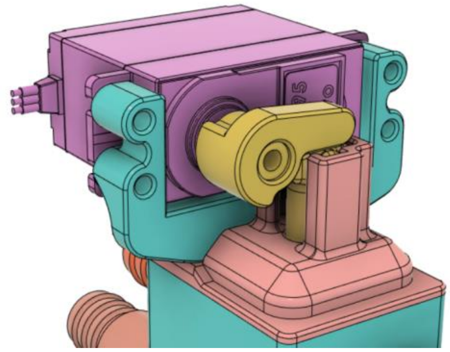

Figure 85: Servo Horn installed to the Servo

9. Glue the Outlet Body Base with the Outlet Body using Silicone Glue

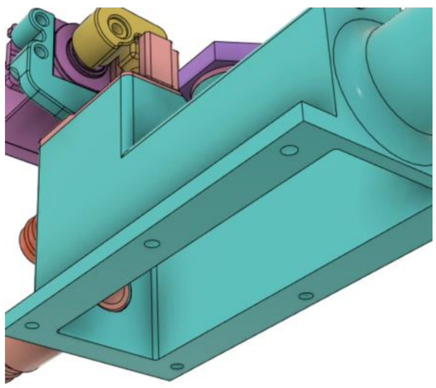

Figure 86: Without Outlet Base

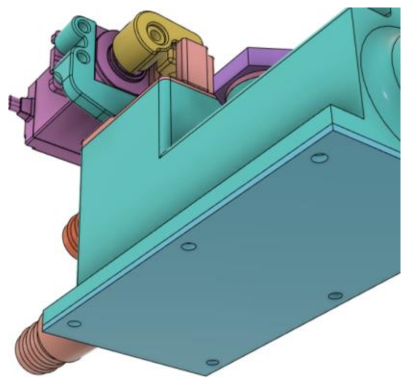

Figure 87: With Outlet Base

2.5 Filter and Pump Assembly

1. Install the HME Inlet Mesh and put the HEPA filter inside. Then screw in the HME Cover onto place.

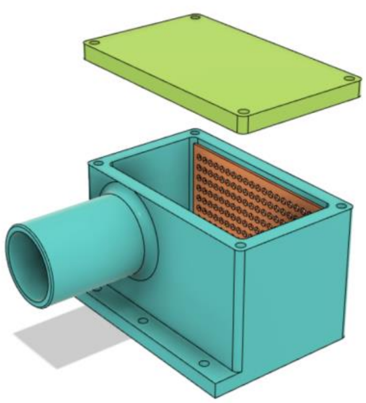

Figure 88: HME Cover open

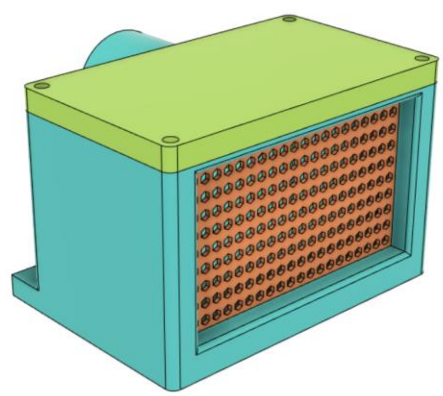

Figure 89: HME Filter assembled

2. Screw down the HME Filter with the Casing Base 12

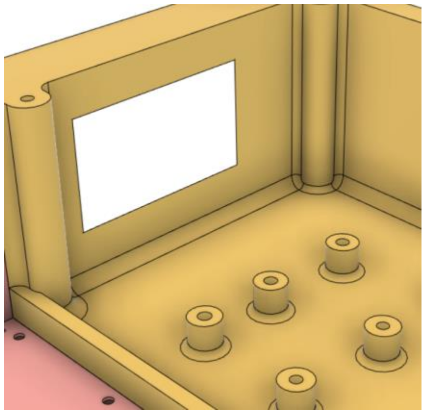

Figure 90: Before screwing down the HME Filter with the Base 12

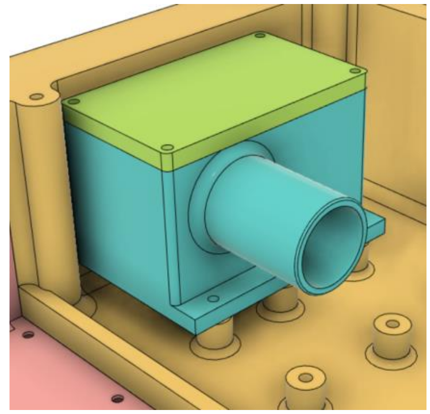

Figure 91: Before screwing down the HME Filter with the Base 12

3. Install the Proportional Valve Body with the Inlet Body. Screw the bodies together from below.

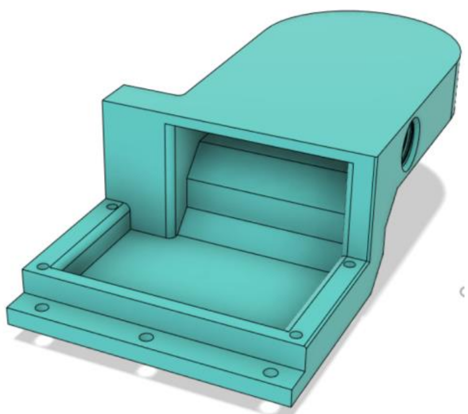

Figure 92: Before screwing down the Proportional Valve Body

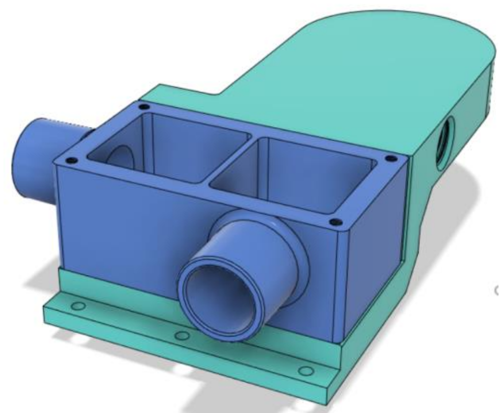

Figure 93: After screwing down the Proportional Valve Body

4. Screw the 9g Micro Servos on the top of the Proportional Valve Cover and then screw the Proportional Valve Cover with the Proportional Valve Body

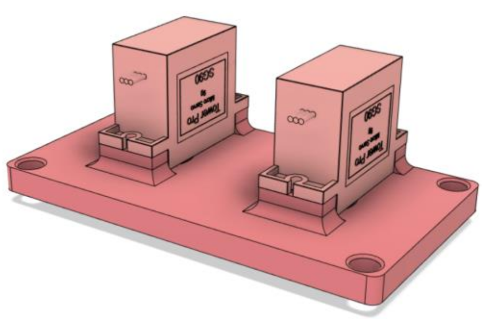

Figure 94: Screw down the Micro Tower 9g servos on the Proportional Valve Cover

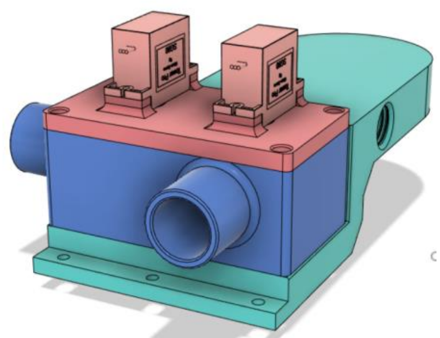

Figure 95: Screw down the Proportional Valve Cover onto the Proportional Valve Body

5. Press-fit the WM7040 Pump to the Inlet Body

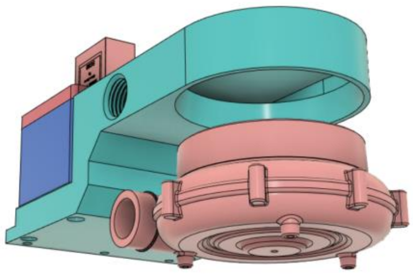

Figure 96: Before press-fitting the WM7040 Pump to the Inlet Body

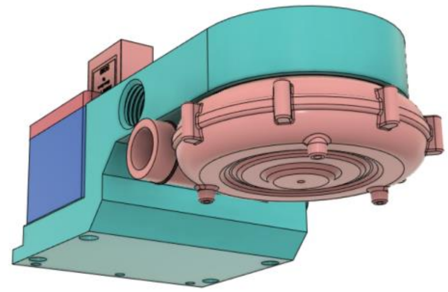

Figure 97: After press-fitting the WM7040 Pump to the Inlet Body

6. Screw in the 14mm Adapter (Short) with the Inlet Body and press-fit the 14mm Adapter (Pump outlet) with the Pump

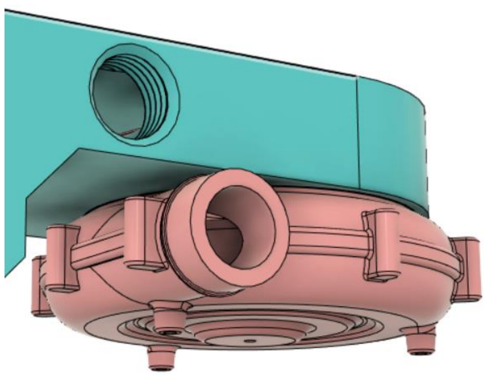

Figure 98: Before installing the 14 mm adapters

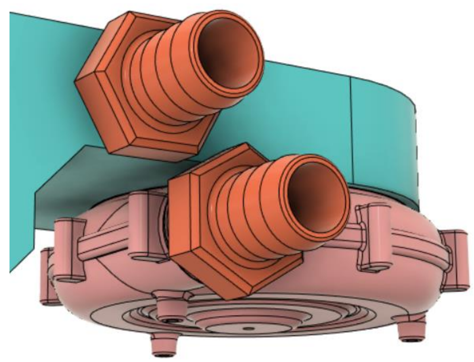

Figure 99: After installing the 14 mm adapters

7. Slide the Inlet Body and Pump assembly with the HME Filter outlet

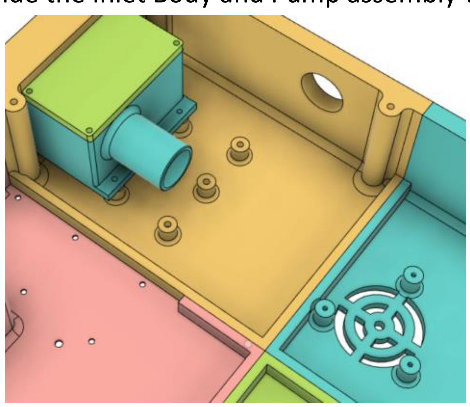

Figure 100: Before sliding in the Inlet Body and Pump assembly

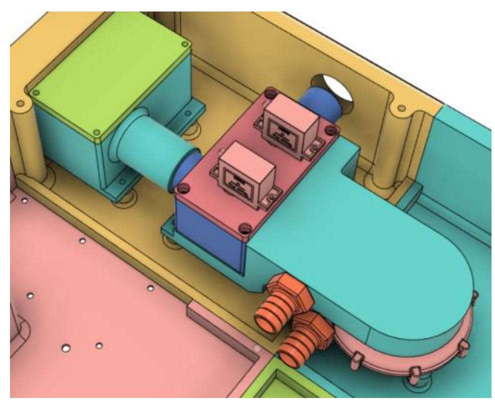

Figure 101: After sliding in the Inlet Body and Pump assembly

8. Screw the 6 screws for installing the Inlet Body and Pump assembly with the Casing Base (12 and 21)
9. Screw the Oxy Nozzle Body with the Oxy Nozzle Holder

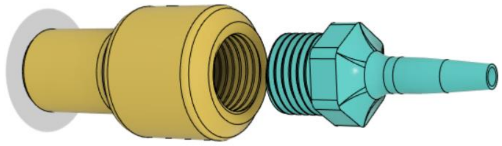

Figure 102: Before screwing the Oxy Nozzle Body with the Oxy Nozzle Holder

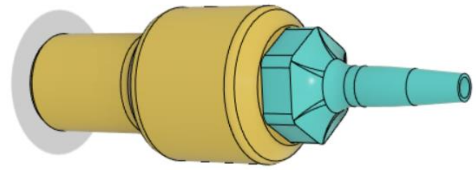

Figure 103: After screwing the Oxy Nozzle Body with the Oxy Nozzle Holder

10. Press-fit the Oxy Nozzle with the Inlet Body

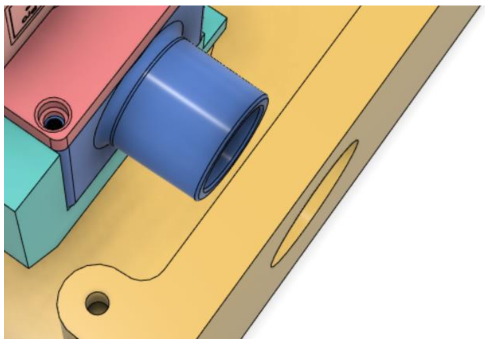

Figure 104: Before press-fiting the Oxy Nozzle Body with the Oxy Nozzle Holder

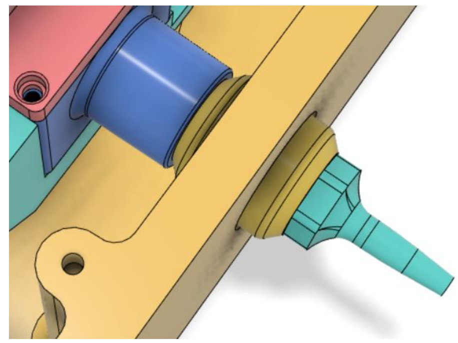

Figure 105: After press-fiting the Oxy Nozzle Body with the Oxy Nozzle Holder

2.6 Final Assembly

1. Screw down the Pressure Release Mechanism (PRM) with the Casing Base 11

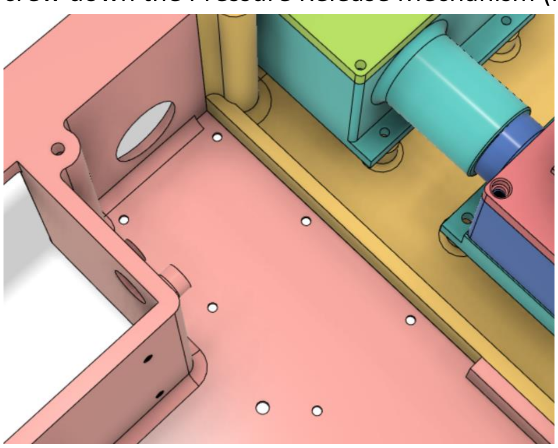

Figure 106: Before screwing down the Pressure Release Mechanism (PRM) with the Casing Base 11

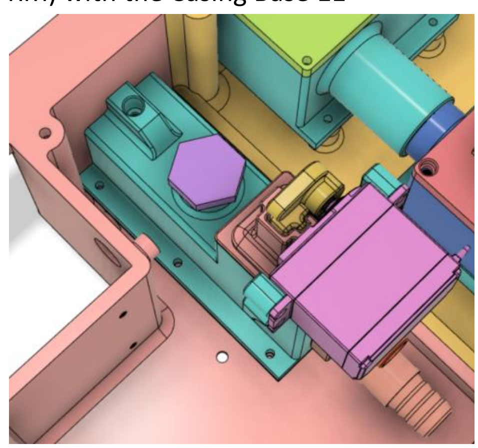

Figure 107: After screwing down the Pressure Release Mechanism (PRM) with the Casing Base 11

2. Attach the One-way Valve to the outlet of the PRM

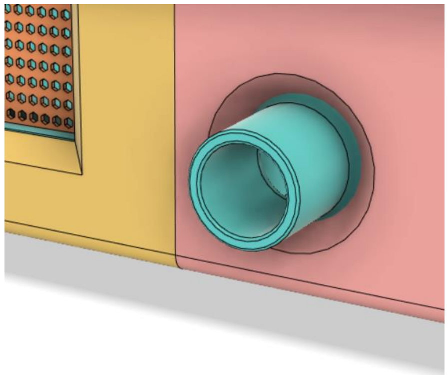

Figure 108: Before attaching the One-way Valve to the outlet of the PRM

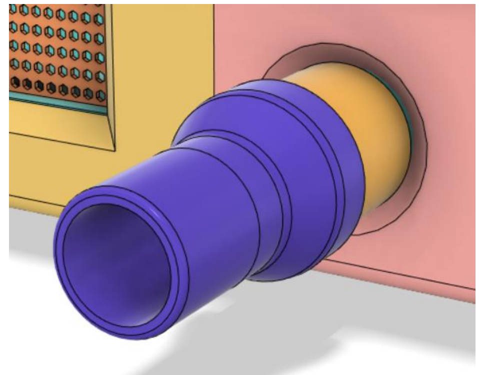

Figure 109: After attaching the One-way Valve to the outlet of the PRM

3. Install Silicone Tube (Hose Release) with the Adapter 14mm (short) of the Inlet Body to the Adapter 14mm (short) of the PRM and install Silicone Tube (Hose Oxy) with the Adapter 14mm (Pump Outlet) of the Pump to the Adapter 14mm (Long) of the PRM

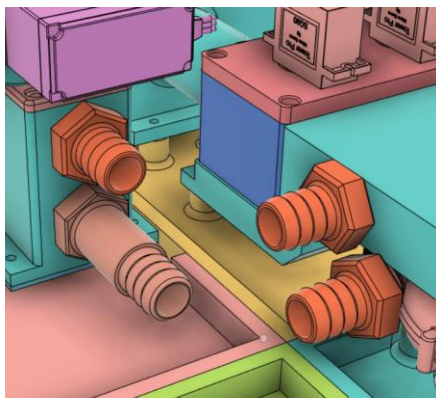

Figure 110: Before attaching the silicone tubes

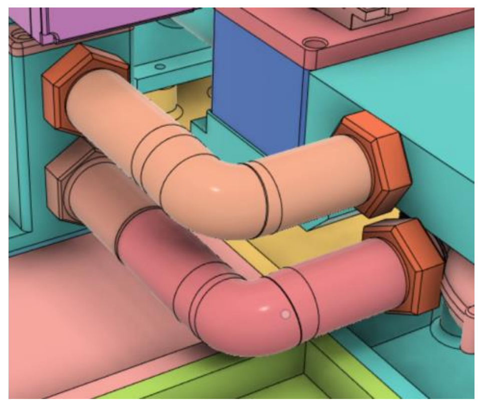

Figure 111: After attaching the silicone tubes

4. Attach the PEEP Hose with PRM's outlet body and put it through Casing Base 11 for it to be attached with the PEEP Valve a bit later

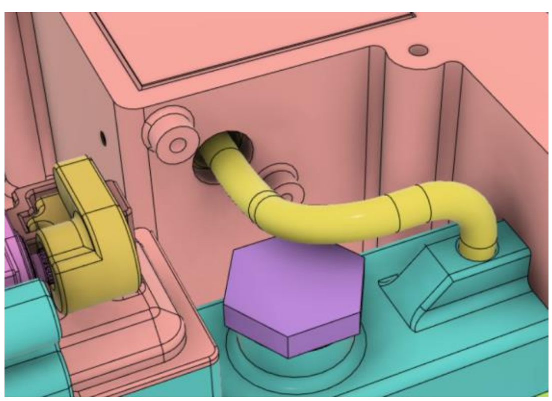

Figure 112: The PEEP Hose with PRM's Outlet Body and putting it through the Casing Base 11

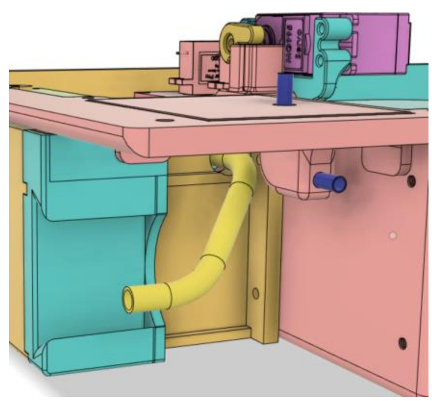

Figure 113: The PEEP Hose through the Front Extension 1 and ready to connected to PEEP Valve later on

5. Slide in the PEEP Valve through the Front Extension 1 and attach the PEEP Hose with it

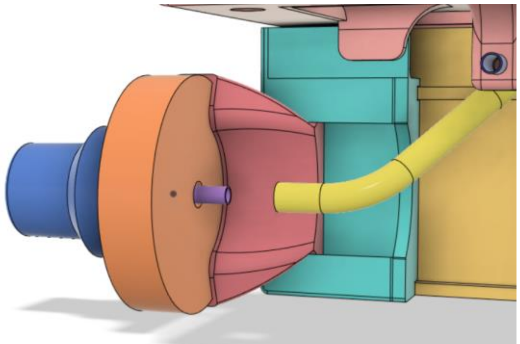

Figure 114: Sliding in the PEEP Valve through the Front Extension 1

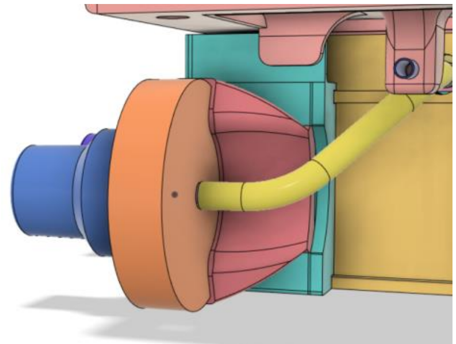

Figure 115: Attached the PEEP Hose with the PEEP Valve

6. Screw the PSU with the PSU Mount and slide into the PSU Holder Tabs to finally screw down the PSU Mount with Casing Base (22 and 11)

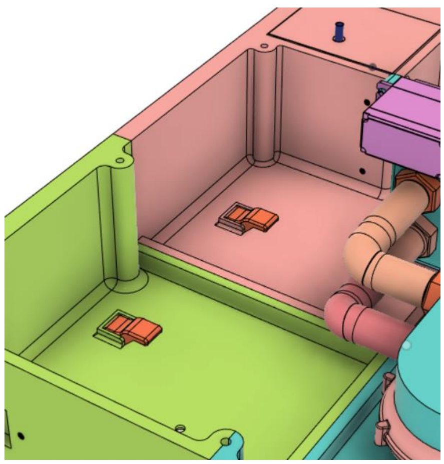

Figure 116: Before installing the PSU Mount and the SMPS

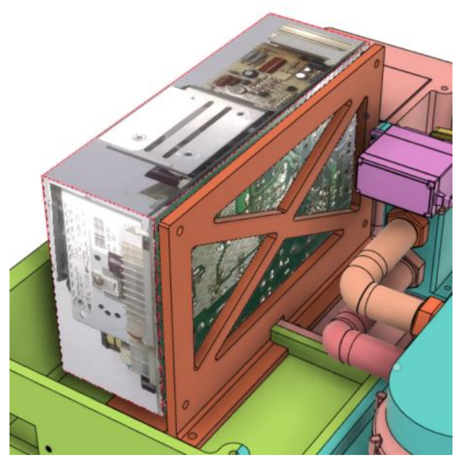

Figure 117: After installing the PSU Mount and the SMPS

7. Glue the Battery Case (Holder) with the Casing Base (22)

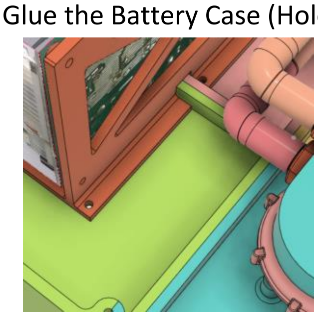

Figure 118: Before gluing the Battery Case (Holder) with the Casing Base (22)

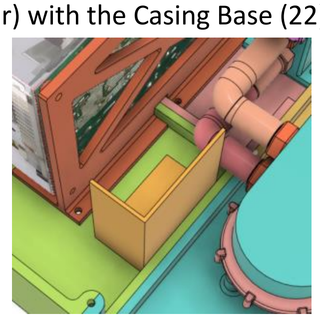

Figure 119: After gluing the Battery Case (Holder) with the Casing Base (22)

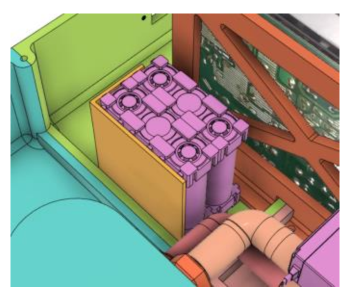

Figure 120: After securing the battery pack with the Battery Case (Holder)

8. Casing Base with installed components

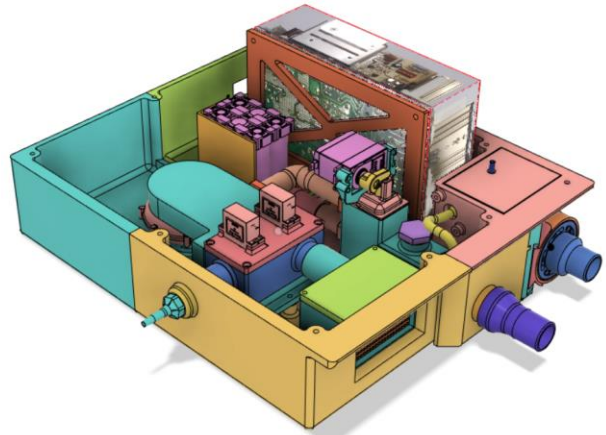

Figure 121: Casing Base with installed components

2.7 Final Assembled Ventilator

Figure 122: Front View 1

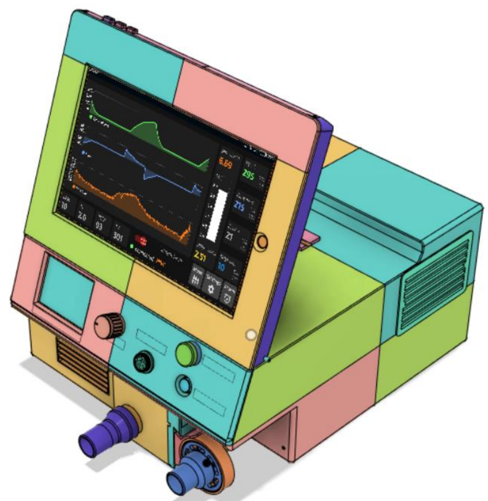

Figure 123: Front View 2

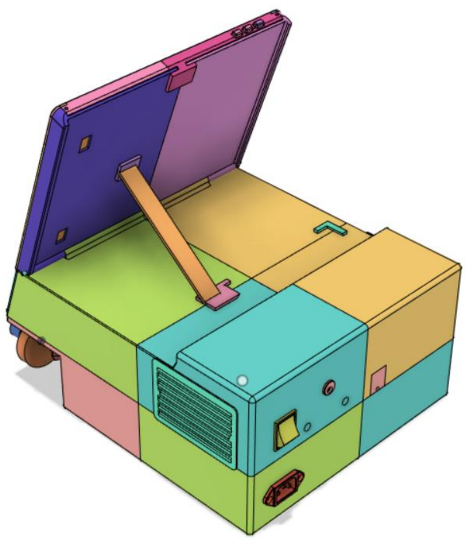

Figure 124: Rear View 1

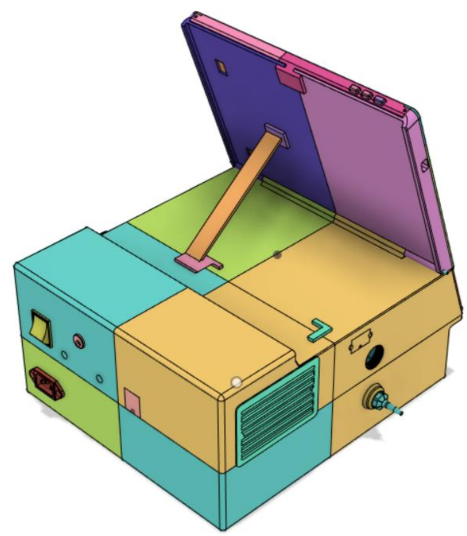

Figure 125: Rear View 2

3D Parts Setup:

1. Open the ‘Assembly 3D Model Files’ and find the ‘Ventilator Assembly.step’ file.
2. Open the ‘Ventilator Assembly.step’ file on Fusion360 or any other 3D modelling software that can open this file.
3. Create STL files for each of the parts in this file, which are also mentioned in the ‘Design Files’ section.
4. Open the STL files in a slicing software and upload it on a 3D printer for printing the parts.
5. After the parts are all printed, follow the detailed assembly instruction provided in the ‘Build Instructions’ section above.
6. Nevertheless, flow sensor’s enclosure can be found in the same folder and it can be produced by following similar steps as mentioned above.

Electronics Setup:

1. Get all the modules as demonstrated through the [Figure 11].
2. Connect the modules accordingly using good quality wires.

Software Setup:

1. Use Android Studio software to generate the apk files.
2. Install into the android device. All files can be found in ‘Android Application’ folder.

Firmware Setup:

1. Download all the codes from the ‘Firmware’ folder
2. Connect an Arduino development board
3. Upload the code onto it

Ventilator Specification:

Summarized specifications for the ventilator are demonstrated in the [Table 3].

Table 3: Ventilator specification

Tidal Volume	100 to 999 ml*
PIP	0 to 35 cm H_2O*
BPM	6 to 60*
PEEP	0 to 12 cm H_2O*
Modes	PRVC, PCV/ BiPap (SIMV available for both)
Alarms	TV, PIP, PEEP, Disconnection of patient circuit, Flow sensor electrical disconnection
Power Supply	13.6V to 24V (20-30W), 100V - 240V AC

*Limits restricted through GUI software but hardware and firmware capacity are much higher

Safety Concerns:

- Under continuous operation for a few years, some of the mechanical and 3D printed parts would need to be changed.
- Prolonged use of filters (HEPA and HME) might decrease the device's performance.

6. Operation Instructions

- Power ON the ventilator.
- Connect Pressure and Flow sensor with the ventilator.
- Connect the Patient circuit and make sure the One-way valve and the PEEP valve is in the right position.
- Turn ON the physical power switch of the ventilator.
- The pressure sensor will be calibrated automatically
- Turn ON the Android Tab/Phone
- Open "CRUX Ventilator Android App"
- Connect the Ventilator
- From the app, screen select the desired parameters. e.g., Mode, Pressure, Tidal Volume, Trigger Flow, etc.)
- From the right bottom corner of the app select Alarm and set desired parameters.
- From the settings option, you can turn ON/OFF debug data (which will be shown in the top left corner)
- In the bottom middle part, there is an ON/OFF switch. Press ON to start ventilation.
- Press the same switch off and a confirmation will be shown. To stop ventilation, press OK.

7. Validation and Characterization:

The main phase of development of the device took place during the first wave of the Covid-19 at the small city of Bangladesh; Sylhet. The development moving as fast as possible and as much as testing possible was conducted, however, due to a lot of barriers such as lack of proper funding and trading bans the procurement of ventilator tester such as, VT650 [22], Citrex H5 [23], etc. could not happen. Hence our device has not been able to be tested through these ventilator testers. Nevertheless, the device was tested in a couple of ways and effective results were achieved which was verified by professional doctors.

a. The analog values from the sensor were verified and converted to cm H_2O pressure as it is more convenient when it comes to implementing different pressure levels. Besides that, doctors are also used to cm H_2O pressure. Hence, the analog values were converted using [Equation 10] and it can be found at [26] under SIMV patient trigger condition [Code: volume.ino >>> 128 no line].

$$float\ mpxCm = (mpxAnalog - mpx2010AnalogOffset) * 0.20916 \text{ --- (10)}$$

Upon not coming across any ready to be used formula for converting analog value to cm H_2O, we fabricated our own setup in finding out the constant; 0.20916, used in the [Equation

10]. A t-connector was connected with a syringe on one end and one of the pressure sensing outlets of MPX2010DP, then the third end was connected with a 4 mm pipe and was put into water held by a measuring cylinder. A ruler was held in parallel to the measuring cylinder by its side by a clamp. The height of the water in cm was recorded and then a pressure was exerted by the syringe. When we saw approximately 1 cm H_2O displacement of water due to pressure exerted by the syringe, the change in analog value of the differential pressure was recorded. This experiment was repeated a couple of times and the mean value was finally taken to calculate the constant that we are finally using in the formula.

b. The experiment was done on the Draeger test lung [27] to simulate real patient conditions. [Figure 126] demonstrates the test lung with the flow sensor. The Draeger test lung is designed and tested for a lot of Draeger ventilators to be operated in any clinical setting. Innovative carbon fiber, silicone, and polysulfone materials are used in its construction; enough to work with the system while delivering the reliable service. This ultra-lightweight test lung merely weighs 190 grams (0.42 pounds). Besides that, it can be completely disassembled for easy cleaning and autoclavable [27].

Figure 126: Draeger test lung with flow sensor connected to the patient circuit

c. SFM 3300 flow sensors [12] were used to verify the magnitude of tidal volume delivered to patients. These sensors are already used in ventilators and other respiratory devices for flow measurements. It is well-known because of its autoclavable feature, capability to measure bi-directional flow rates of 250 slm and small dead space of less than 10 ml. Moreover, the biggest advantage of using it is that it has a very fast update time (0.5 ms) and calibration is not needed.

Two of such flow sensors were connected in series with the patient circuit through which the air flowed from the turbine to the patient. The first one was connected to the microcontroller directly and data was retrieved using the serial monitor on a computer. The second one was connected to another computer through the official interfacing tool provided by Sensiron EK-F3x-CAP [28]. Data was compared and almost no differences were found amongst the data retrieved by both of the setups.

d. Efficiency is calculated using the values obtained through two methods; J7-t power tested [29] and DPS5005 power supply [30].

J7-t tester has very low power consumption (20mA current consumption) by itself while displaying the values. Displayed voltages and currents have accuracy of 0.03V and 0.02A respectively; comparatively higher accuracy [29].

DPS5005 power supply has a resolution of 0.01 for both voltages and currents measured with 100mV(peak to peak) at max workload [30].

Finally, graphs were obtained [Figure 127] and [Figure 128] showing the operation of the device in PRVC and PCV modes respectively.

Figure 127: Screenshot of PRVC mode’s GUI

Figure 128: Screenshot of PCV mode’s GUI

8. Acknowledgements

The authors feel blessed to have so many supports along the path of our development and wish to acknowledge the following supports:

- North East Medical College Hospital - Demonstrated ventilator operations which helped in the understanding of the system and also donating a scraped ventilator for our development
- Mr. Aminul Islam Chondon, Electro-Medical Technologist - showed the internal hardware of an old ventilator
- Cybernetics Robo Ltd - for providing an office space and some tools for the team throughout the development
- Dr. Sabyasachi Roy, Parkview Medical College, Sylhet - valuable thoughts about ventilator modes, which eventually led to effective algorithm design of the system
- Syed Aminur Rahman - Providing 3D printing material to print our 3D models
- Dr. Akhter Uddin Murad, Founder, Akhter Medical & Research Institute- for providing hospital facility & medical tools for development works
- Dr. Asef Bakhtiar Chowdhury: for demonstrating ventilator and BiPap machine in ICU condition

9. Conclusion

Technology is always developing around the world but there is still some gap in the implementation of intensive care medical devices in finding ways to reduce cost while keeping the functionalities the same if not better. We have successfully developed a power efficient system with two completely novel designs; PRM and PEEP valves. These valves have enabled the system to achieve more than 35 times efficiency running along-side effective control algorithms such as, feed-forward and PID. The modular low cost and portable system can easily be deployed.

However, we do acknowledge that there are still some developments to be done and these are already in the process; such as, more modes are to be developed, the oxygen blending method needs to be corrected and finally the primary display system needs to be developed with a simpler display that would be able to provide absolutely minimal and crucial information so that this system can even run at the scenario where an android tablet is not available.

10. Declaration of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References:

[1] Raspberrypi.org. 2021. [online] Available at: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/> [Accessed 6 August 2021].

[2] "Radxa Wiki", Wiki.radxa.com, 2014. [Online]. Available: https://wiki.radxa.com/News/2014/10/rock-pro-lite-is-oshw-now.

[3] T. Heffernan, "The Best Air Purifier", Wirecutter: Reviews for the Real World, 2021. [Online]. Available: https://www.nytimes.com/wirecutter/reviews/best-air-purifier/

[4] C. Dugdale and R. Walensky, "Filtration Efficiency, Effectiveness, and Availability of N95 Face Masks for COVID-19 Prevention", JAMA Internal Medicine, vol. 180, no. 12, p. 1612, 2020. Available: 10.1001/jamainternmed.2020.4218.

[5] "Filter/ HME", *Draeger*, 2021. [Online]. Available: https://www.draeger.com/en_in/Products/Filter-and-Heat-and-moisture-exchanger [Accessed: 06- Aug- 2021].

[6] "ventilator centrifugal blower fan DC12V 24V air blower ventilator medical respirator centrifugal fan Inflatable ventilateur 7040|Blowers| - AliExpress", aliexpress.com, 2021. [Online]. Available: https://www.aliexpress.com/item/10000048326249.html?spm=a2g0s.8937460.0.0.438c2e0ePyDUwe [Accessed: 06- Aug- 2021].

[7] "MG995 Servo Motor", Components101, 2021. [Online]. Available: https://components101.com/motors/mg995-servo-motor.

[8] P. Acosta, E. Santisbon and J. Varon, "“The Use of Positive End-Expiratory Pressure in Mechanical Ventilation”", Critical Care Clinics, vol. 23, no. 2, pp. 251-261, 2007. Available: 10.1016/j.ccc.2006.12.012.

[9] "Arduino Mega 2560 Rev3 | Arduino Official Store", Store.arduino.cc, 2021. [Online]. Available: https://store.arduino.cc/usa/mega-2560-r3.

[10] "Bluetooth Module HC-05 | Sensors & Modules", Electronicwings.com, 2021. [Online]. Available: https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-.

[11] "MPX2010 Series", Nxp.com, 2021. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/MPX2010.pdf.

[12] "Digital Flow Meter for medical applications", Mouser.com, 2021. [Online]. Available: https://www.mouser.com/datasheet/2/682/Sensirion_Mass_Flow_Meters_SFM3300_Datasheet-1524535.pdf.

[13] "Lithium Ion NCR18650B", Batteryspace.com, 2021. [Online]. Available: https://www.batteryspace.com/prod-specs/NCR18650B.pdf.

[14] P. Rodriguez, M. Dojat and L. Brochard, "Mechanical ventilation: changing concepts", ResearchGate, 2021. [Online]. Available: https://www.researchgate.net/publication/278001931_Review_Article_-_Mechanical_ventilation_changing_concepts.

[15] "Practical differences between pressure and volume controlled ventilation | Deranged Physiology", Derangedphysiology.com, 2021. [Online]. Available: https://derangedphysiology.com/main/cicm-primary-exam/required-reading/respiratory-

system/Chapter 542/practical-differences-between-pressure-and-volume-controlled-ventilation.

[16] "Pressure Control Ventilation", Weinmann-emergency.com. [Online]. Available: https://www.weinmann-emergency.com/solutions/ventilation-modes/pressure-control/.

[17] A. Carpio and J. Mora, "Ventilator Management", Ncbi.nlm.nih.gov, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK448186/.

[18] "Multithreading in Java - javatpoint", www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/multithreading-in-java.

[19] "Asynchronous | What is Asynchronous - javatpoint", www.javatpoint.com, 2021. [Online]. Available: https://www.javatpoint.com/asynchronous.

[20] "Handler | Android Developers", Android Developers. [Online]. Available: https://developer.android.com/reference/android/os/Handler. [Accessed: 07- Aug- 2021].

[21] "BluetoothSocket | Android Developers", Android Developers. [Online]. Available: https://developer.android.com/reference/android/bluetooth/BluetoothSocket. [Accessed: 07- Aug- 2021].

[22] "string::substr - C++ Reference", Cplusplus.com. [Online]. Available: https://www.cplusplus.com/reference/string/string/substr/. [Accessed: 07- Aug- 2021].

[23] R. Chatburn, M. El-Khatib and E. Mireles-Cabodevila, "A Taxonomy for Mechanical Ventilation: 10 Fundamental Maxims", Respiratory Care, vol. 59, no. 11, pp. 1747-1763, 2014. Available: https://www.researchgate.net/publication/264796263_A_Taxonomy_for_Mechanical_Ventilation_10_Fundamental_Maxims.

[24] "VT650 Gas Flow Analyzer | Fluke Biomedical", Flukebiomedical.com. [Online]. Available: https://www.flukebiomedical.com/products/biomedical-test-equipment/gas-flow-analyzers-ventilator-testers/vt650-gas-flow-analyzer-ventilator-tester.

[25] "Citrex H5 | Ventilator Testing | Rigel Medical", Rigel Medical. [Online]. Available: https://www.rigelmedical.com/gb/products/ventilator-testing/gas-flow-analyzers/800a9003-citrex-h5/.

[26] S. Haque, "ventilator/volume.ino at main · Nabilphysics/ventilator", GitHub, 2021. [Online]. Available: https://github.com/Nabilphysics/ventilator/blob/main/vent_SIMV_1.0/volume.ino. [Accessed: 07- Aug- 2021].

[27] Dräger SelfTestLung™. Draeger, 2021, pp. 1-4.

[28] "Quick Start Guide EK-F3x-CAP", Sensirion.com, 2021. [Online]. Available: https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/5_Mass_Flow_Meters/Sensirion_Mass_Flow_Meters_EK-F3x-CAP_Quick_Start_Kit_EN.pdf.

[29] "Test of USB safety tester J7-t", Lygte-info.dk, 2021. [Online]. Available: https://lygte-info.dk/review/USBmeter%20safety%20tester%20J7-t%20UK.html. [Accessed: 07- Aug- 2021].

[30] "DPS5005 - Joy-IT", Joy-it.net, 2021. [Online]. Available: https://joy-it.net/en/products/JT-DPS5005. [Accessed: 07- Aug- 2021].