

**Title:** VentMon: and Open Source Inline Ventilator Test Fixture and Monitor

**Authors:** Robert L. Read, Lauria Clarke, Geoff Mulligan

**Affiliations:** Public Invention with support from the *Protocol Labs* and the *Mozilla Foundation*

**Contact email:** read.robert@gmail.com

**Abstract:**

Humanitarian engineers responded to the pandemic ventilator shortage of March, 2020, by beginning over 100 open source ventilator projecta[6]. Inexpensive ventilator test equipment facilitates projects forced to be geographically distributed by lockdowns. The VentMon is a modular, open source, IoT-enabled tester that plugs into a standard 22mm airway between a ventilator and a physical test lung to test any ventilator by graphing flow, pressure, oxygen fraction, humidity and temperature at a data lake accessible to all team members. This is the data needed by a test team, but also the data desired by a clinician in actual treatment of patients. The open source design of the VentMon and its firmware and cloud-based software may allow it to be used as a component of modular ventilators to provide a clinical readout. This software system has been designed to be as modular and composeable as possible, for example by defining new open published standards, just as the hardware is composeable and modifiable, forming an open system or eco-system of ventilation data. Thanks to grants, 20 VentMons have been given away free of charge to pandemic response teams building open source ventilators, some of which have been used successfully.

**Keywords:** COVID-19, open source medical device, ventilator, pandemic response, IoT, respiration waveform

## Specifications table:

<b>Hardware name</b>	VentMon
<b>Subject area</b>	Educational Tools and Open Source Alternatives to Existing Infrastructure
<b>Hardware type</b>	Field measurements and sensors, Electrical engineering and computer science
<b>Open source license</b>	MIT License, CERN-OHL-S
<b>Cost of hardware</b>	\$280
<b>Source file repository</b>	<p>VentMon: <a href="https://doi.org/10.5281/zenodo.4079170">https://doi.org/10.5281/zenodo.4079170</a></p> <p>PIRDS (Public Invention Respiration Data Standard): <a href="http://doi.org/10.5281/zenodo.4079377">http://doi.org/10.5281/zenodo.4079377</a></p> <p>PIRDS logger: Zenodo. <a href="http://doi.org/10.5281/zenodo.4079382">http://doi.org/10.5281/zenodo.4079382</a></p> <p>SFM3X00: <a href="http://doi.org/10.5281/zenodo.4079355">http://doi.org/10.5281/zenodo.4079355</a></p> <p>VentDisplay: <a href="http://doi.org/10.5281/zenodo.4079374">http://doi.org/10.5281/zenodo.4079374</a></p> <p><i>DOI URL to an approved source file repository: Mendeley Data, the OSF, or Zenodo (instructions). For example:</i> <a href="https://doi.org/10.5281/zenodo.3346799">https://doi.org/10.5281/zenodo.3346799</a></p>

### 1. Hardware in context

The Coronavirus pandemic has created a potential global shortage of ventilators. Ventilators are expensive to manufacture and during uncertain times supply chains can be disrupted, increasing the cost and scarcity of these devices. Since the beginning of the pandemic in March, there has been a large movement to develop cheaper, more supply chain-resilient, ventilators. The urgency and goals of this movement have shifted as the global understanding of the virus has evolved. Regardless of the current state of this movement, however, the use of open source medical devices in resources limited emergency situations—common during a global pandemic—is a topic that has come to the forefront of many conversations about preparedness and treatment. Medical devices require rigorous testing before they can be used on the general public. Open source medical devices require the same level, if not more scrutiny. If an open source movement is to succeed in this field, the tools to test and validate medical devices should also be community based. The goal of VentMon is to provide an equally open source, community based verification solution to increase the efficiency and accuracy of the development process for socially distanced, potentially international teams making open source ventilator, BPAP, and PAPR devices.

## 2. Hardware description

The VentMon plugs directly into a standard 22mm adult airway. As air or medical gases pass through the VentMon, it records flow and pressure about 40 times a second, and occasionally records temperature, humidity, and O<sub>2</sub> fraction. The result is that when placed in the airway between an invasive or non-invasive ventilator under test and physical test lung, it sends this data to a public data lake [4] using a clearly defined respiration standard. This data can be graphically rendered either live or statically to provide a display that is typical of the clinical display of advanced ventilators. In this way it allows sophisticated testing of ventilators. This device is therefore similar in function to commercial Test Lung devices, which tend to start at USD\$10,000.

A special challenge of teams building open source pandemic ventilators is that they tend to be poorly funded, but also geographically distributed. The VentMon is an IoT-enabled device which sends its data to a public data lake, where a team member on a different continent can view a waveform live in real time as a colleague makes adjustments to the ventilator.

The VentMon thus offers benefits to teams developing ventilation devices:

- It is inexpensive.
- It is completely open and part of a modular hardware/software system that can be customized and extended.
- It allows remote collaboration.

Moreover, being completely open, it supports modification, extension, and importantly but only potentially at present, integration into a complete ventilator. A ventilator that chose to offer a graphical trace of pressure and flow, which are generally demanded by clinicians for all but emergency transport situations, could incorporate the open source VentMon into its complete design.

Although the reliance on the internet is a strength for a distributed engineering team, it would be a disadvantage in a field hospital. Although aiming to eventually be safe enough and reliable to be used as a monitor on human patients, it requires much improvement before it is ready for that intended use.

### 3. Design files

#### 3.1 Design Files Summary

Design file-name	File type	Open source license [check these]	Location of the file
(1) Embedded Firmware	C++ Source Code	MIT License	<a href="https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/releases/tag/v0.2">https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/releases/tag/v0.2</a>
(2) Embedded Firmware (v0.2 COTS version)	C++ Source Code	MIT License	<a href="https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/tree/master/VentMonFirmware">https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/tree/master/VentMonFirmware</a>
(3) PIRDS Data Viewer (Vent Display)	HTML Source Code	MIT License	<a href="https://github.com/PubInv/vent-display">https://github.com/PubInv/vent-display</a>
(4) PIRDS Data Logger	C Source Files	MIT License	<a href="https://github.com/PubInv/PIRDS-logger">https://github.com/PubInv/PIRDS-logger</a>
(5) PIRDS Data Standard	Documented standard, and C-language binding	Creative Commons Zero (Universal) for documents, GPL 3.0 for code binding	<a href="https://github.com/PubInv/PIRDS-respiration-data-standard">https://github.com/PubInv/PIRDS-respiration-data-standard</a>
(6) VentMon T0.3 PCB	PCB Design Files	MIT License	<a href="https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/tree/master/design/pcb">https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/tree/master/design/pcb</a>
(7) Pressure Sensor and Airway Adaptor	STL File	MIT License	VentMon Adaptor <a href="https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/tree/master/design/3dparts">https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/tree/master/design/3dparts</a>

We promote an open, modular, and composable respiration ecosystem. Therefore, we have more separate repositories and modules than would strictly be needed if we were only making the VentMon project, in hopes that smaller modules may be reusable even by those who have no use for the VentMon specifically.

1. The main firmware is written in the Arduino environment, and though targeting an ESP32 could easily apply to other boards. This core firmware manages reading the various sensors, and outputs its data both on its small built-in OLED (for function testing), and, more importantly, via the PIRDS standard to a public data lake where a separate module renders it with a display appropriate for a clinician.
2. Although our current focus is on a printed circuit board version of the VentMon[1], we report here a system that can be made by an amateur using only commercial off-the-shelf (COTS) parts. This is version T0.2 of the tester, and software that matches it (slightly incompatible with the current version) is tagged.
3. Although the VentMon has a tiny OLED screen and graphs pressure, it is too small for

more than “smoke” testing. For serious performance testing and eventual clinical use, we use VentDisplay, a browser-delivered JavaScript code that presents a full clinical display.

4. The VentDisplay software can be pointed at a logger that serves up data in the PIRDS format. This logger takes data from UDP packages and writes them to a file, which can then be served up by a webserver to any requester.
5. Integral the VentDisplay and PIRDS Data Logger is the Public Invention Respiration Data Standard (PIRDS). This is a documented standard.
6. The VentMon T0.3 printed circuit board is designed with EAGLE CAD.
7. A small 3D printed housing epoxied over the ambient air quality sensor with a tiny pressure port allows us to share the air with the main pressure hose.

*For each design file listed above, include a short description of the file here (one or two sentences)*

#### **4. Bill of materials**

**Current Link:** [VentMon HardwareX BOM](#)

- BOM needs to be converted to PDF and uploaded per instructions:

*For a complex Bill of Materials, the complete Bill of Materials (editable spreadsheet file e.g., ODS file type or PDF file) can be uploaded in an open access online location such as the Open Science Frameworks repository. Include the link here. Alternatively, the Bill of Materials can be uploaded at the time of submission on the online Elsevier submission interface as supplementary material.*

#### **5. Build instructions**

Two versions of VentMon can be assembled depending on availability of parts and time. The most physically robust and complete version of VentMon requires the purchase and manufacture of a PCB as well as a number of 3D printed plastic parts. The designs of the PCB and 3D printed parts are fully open; in practice, due to the fact that PCBs have a high set up cost, it makes most sense to use the PCB if you are building 10 or more VentMons. This version of the device requires the least amount of time to assemble and contains fewest discrete components. Due to the relatively high cost of PCB manufacture and assembly, VentMon can also be created using off the shelf components readily available from DIY electronics suppliers. This version requires significantly more assembly time. We believe this has been done twice by third parties with little interaction with us. Both assembly procedures are outlined below.

##### **5.1 COTS Based VentMon (T0.2)**

*The VentMon T0.2 is not meant to be used on human patients.* The COTS version uses firmware v0.2 tagged as such[5]. This version is somewhat incompatible with v0.3 (described below), which uses a PCB, and which is the focus of our main development.

###### **1. Qwiic Shield Assembly**

See Figure 1 on page 6 for final assembly of Qwiic Shield.

- 1.1 Trim two pieces of male header pins that are two positions long and one piece that is four positions long.
- 1.2 With the connector side of the board facing up, solder one of the two position headers into the two holes labelled SDA and SCL.
- 1.3 Solder the other two position header into the last two holes on the row along the opposite side.
- 1.4 Solder the four position header into the last four holes along the right side. This should provide connections to the GND and 3V3 through holes.
- 1.5 Connect both sides of the solder bridge to engage the pullup resistors for the I<sup>2</sup>C bus.

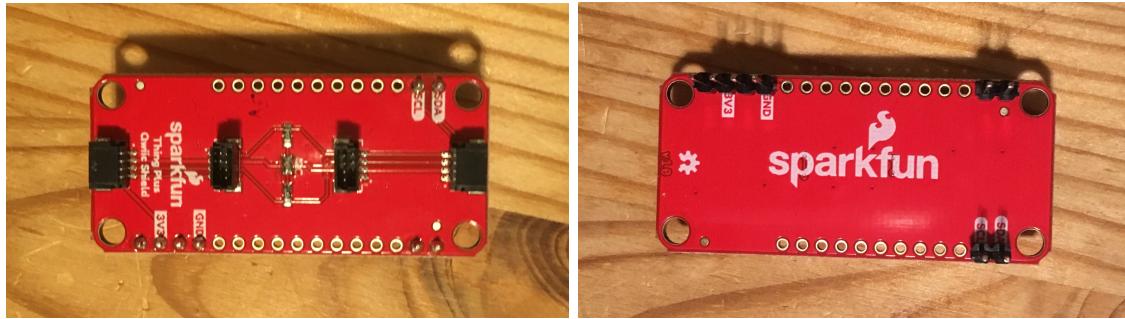


Figure 1: SparkFun Qwiic Shield with male headers and I<sup>2</sup>C pullups connected

## 2. Enclosure

Before starting this step download and print the drill template.

- 2.1 Cut out the template and tape it to the enclosure.
- 2.2 Make sure the lid of the enclosure is firmly screwed on before drilling
- 2.3 Drill holes in the locations denoted on template using the recommended drill sizes for each hole.

## 3. Pressure Sensor Assembly

- 3.1 Drill out mounting holes refer to figure 3.2 on page 7
  - 3.1.1 Using the 7/64" drill bit, carefully enlarge the mounting holes in each corner of the sensor
  - 3.1.2 You can do all four or just two - one on the left side of one board and one on the right side of the other
- 3.2 Cut trace from SDO through hole to sensor. Refer to figure 3.2 on page 7.
  - 3.2.1 Carefully cut the trace coming from the SDO hole on the back side of one board
  - 3.2.2 This board will be used to measure the airway pressure



Figure 2: BME 280 Eval board with drilled out holes and trace cut

### 3.3 Assemble Cable. Refer to figure 3 on page 7.

3.3.1 Cut one connector off the Qwiic cable

3.3.2 Strip and tin the last 1/4" of the leads



Figure 3: Prepared Qwiic cable for BME sensor assembly

### 3.4 Connect Sensors to Cable. Refer to figure 3.4 on page 8 and table 1 on page 8 for electrical connection.

3.4.1 Stack the two sensors on the breadboard and use two paper clips to anchor them

3.4.2 Be sure the airway board with the cut trace is facing down

3.4.3 Starting with Vin or GND insert the tinned lead through the appropriate hole and apply solder

3.4.4 Flip the sensor assembly over and make sure solder has flowed through both through holes, adding some if needed

3.4.5 Don't worry if the solder does not flow perfectly

3.4.6 Pin the assembly back down and move to another connection following the connection table below

3.4.7 Repeat the steps above until the two boards have been secured and all leads are connected

3.4.8 Re-flow all the joints to ensure a good connection

Signal	Qwiic Color
GND	black
VIN	red
SDA	blue
SCL	yellow

Table 1: pressure sensor wiring

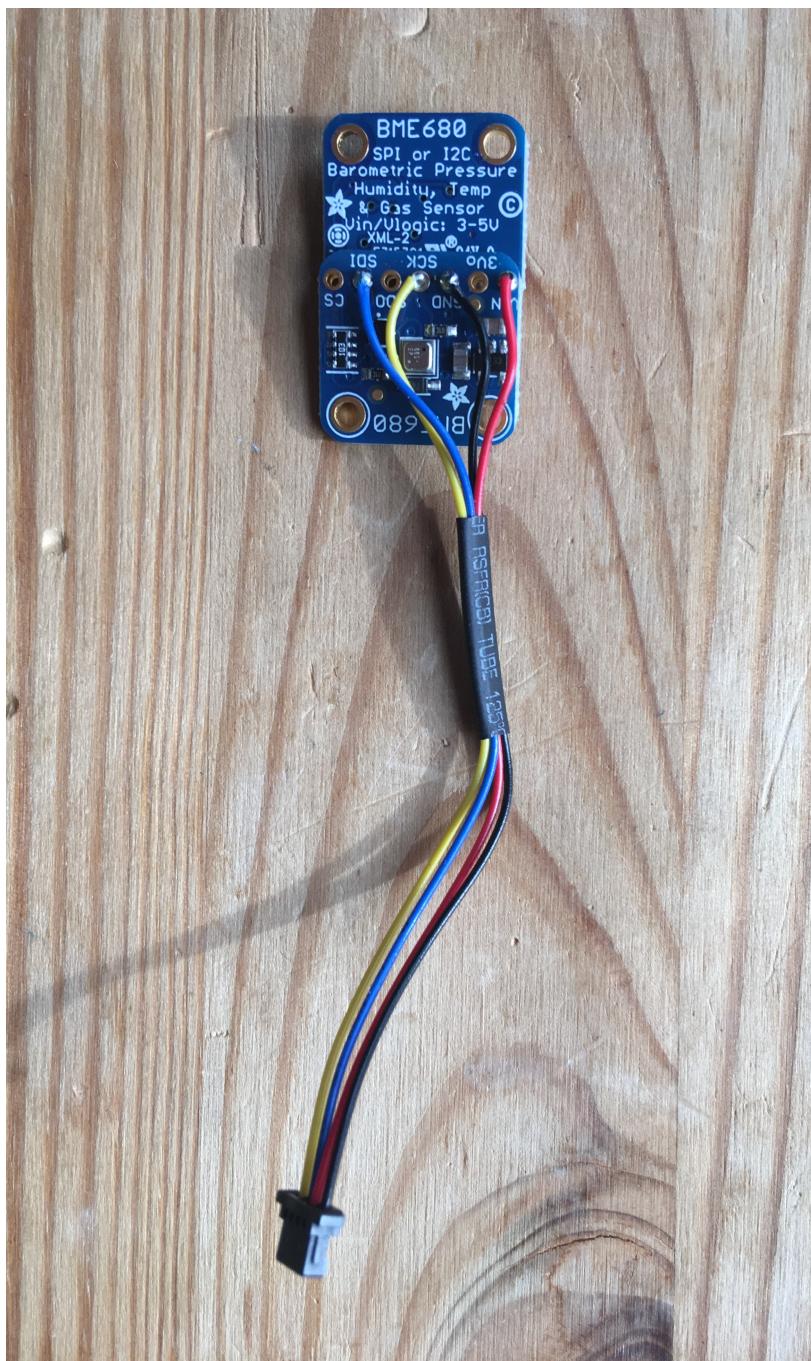


Figure 4: Qwiic cable soldered to BME sensor assembly

- 3.5 Add address jumper to ambient pressure sensor. Refer to figure 5 on page 9
- 3.5.1 Take a 1/2" piece of hookup wire and strip both ends
  - 3.5.2 Flow the solder in the GND through hole and poke one end of the hook up wire into the hole
  - 3.5.3 Put the other end of the wire into the SDO through hole and apply solder

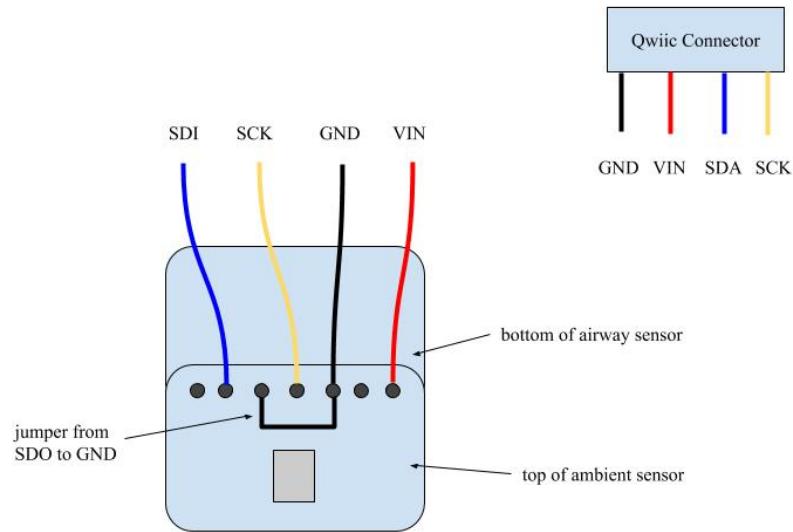


Figure 5: Qwiic cable soldered to BME sensor assembly

## 4. Flow Sensor Assembly

### 4.1 Assemble Cable. Refer to figure 6 on page 10

The flow sensor requires a 5 V input, however we will run the I<sup>2</sup>C bus at 3.3 V necessitating this cable.

- 4.1.1 Cut the Qwiic cable in half
- 4.1.2 Cut off the red conductor near the base of the connector
- 4.1.3 Cut a 30 cm length from the USB cable
- 4.1.4 Strip the jacket back 1 1/2" from one end
- 4.1.5 Strip the first 1/4" of each conductor on one end and tin
- 4.1.6 Twist a small portion of the shield into the black GND conductor
- 4.1.7 Place a small piece of heat shrink on each conductor of the Qwicc cable and push them toward the connector end
- 4.1.8 Solder the ends of the Qwiic cable to the ends of the USB cable following Table 2.

USB	Qwiic	Signal
black	black	GND
green	blue	SDA
white	yellow	SCK
red	single 6" wire	5V

Table 2: flow sensor wiring

4.1.9 Connect the red wire to the red conductor in the USB cable and solder the extra long male header pin to the end

4.1.10 Cover the finished joints with a short 1/4" piece of heat shrink

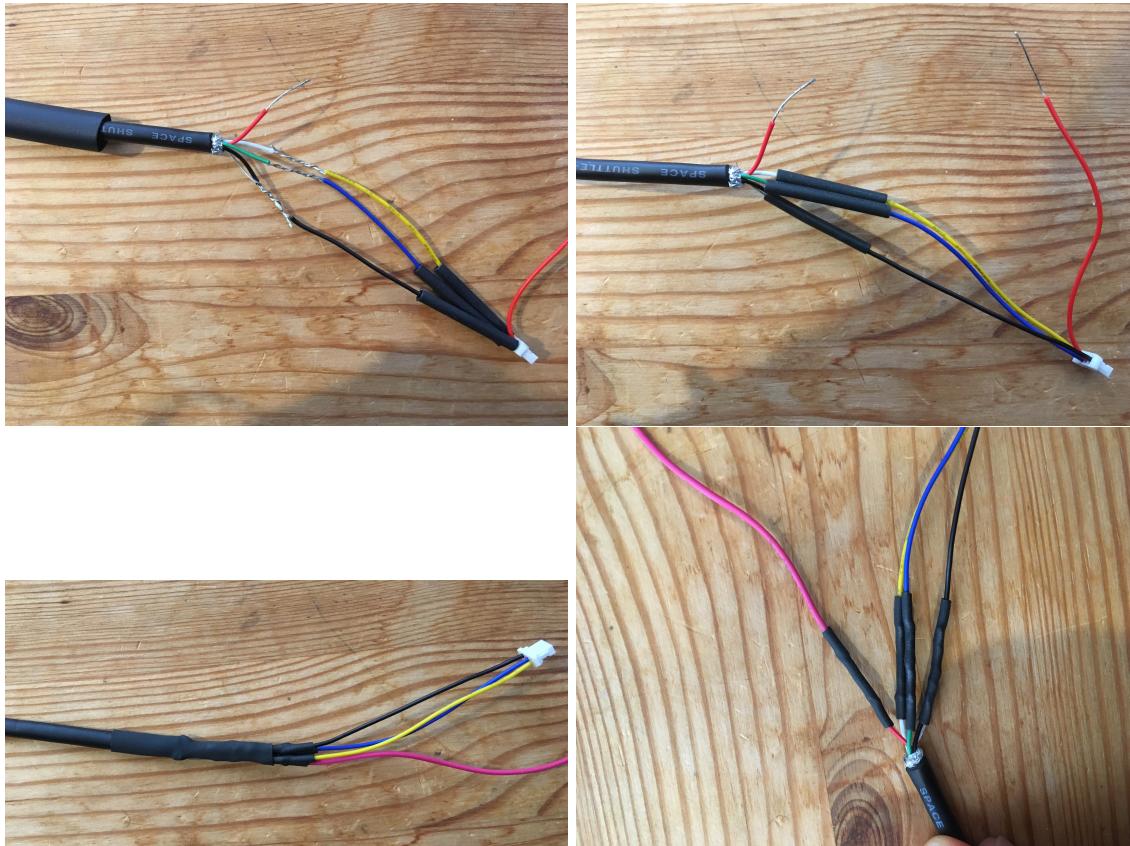


Figure 6: Clockwise progression of sensor cable assembly.

4.2 Connect cable to sensor. Refer to figure 4.2 on page 11.

4.2.1 Apply a small quantity of solder to the four center pads on the SFM3400 flow sensor

4.2.2 Remove 5/8" of the jacket on the bare end of the USB cable

4.2.3 Strip back 1/8" from each conductor and tin the ends

4.2.4 Thread the cable through the hole in the sensor as shown in the photo and make sure the tinned leads can reach the sensor pads

4.2.5 Connect the end of each lead to the pads you have prepared by touching the tip of the soldering iron to the pad and then placing the tinned lead onto it.

4.2.6 Refer to the SFM3400 datasheet to be sure you are connecting them in the proper position.

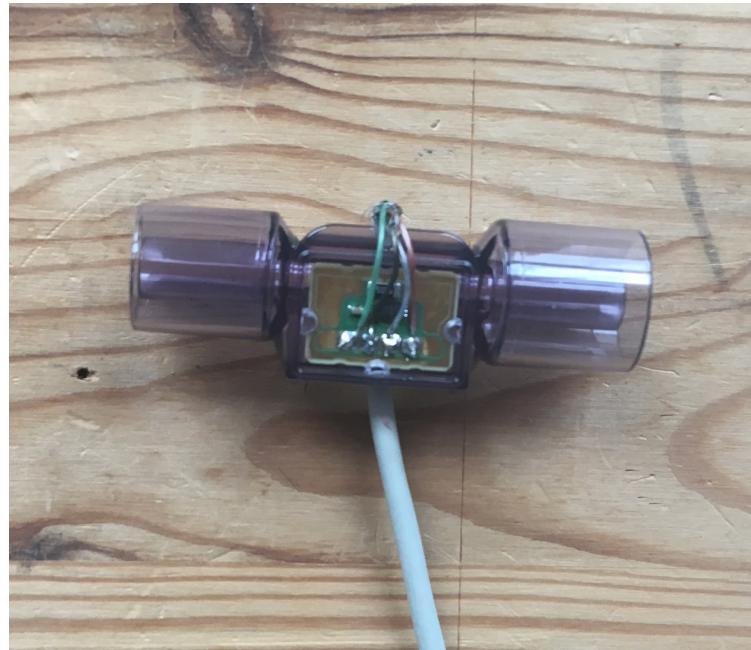


Figure 7: Sensor connected to cable assembly.

4.3 Assemble sensor adapters. Refer to figure 8 on page 12.

- 4.3.1 Cut a 1 1/2" section of petex tubing
- 4.3.2 Insert the tubing into the wide end of the flow sensor
- 4.3.3 Insert the frosted 15 mm to 22 mm adapter over the petex
- 4.3.4 Add the clear 15 mm to 22 mm adapter over the small end of the flow sensor
- 4.3.5 Place the bleed adapter over the clear adapter
- 4.3.6 Use electrical tape to secure any loose connections



Figure 8: Adapters assembled inline with sensor.

## 5. Final Assembly

5.1 Glue oxygen hose to pressure sensor assembly. Refer to figure 9 on page 13.

5.1.1 Cut a section of oxygen hose approximately the same length as the flow sensor cable

5.1.2 Make sure your cut is flat on the end

5.1.3 Place one end of the hose over the airway pressure sensor and apply glue around the edges

5.1.4 Be very careful not to get glue on the sensor and make sure that the hose is securely attached to the pcb

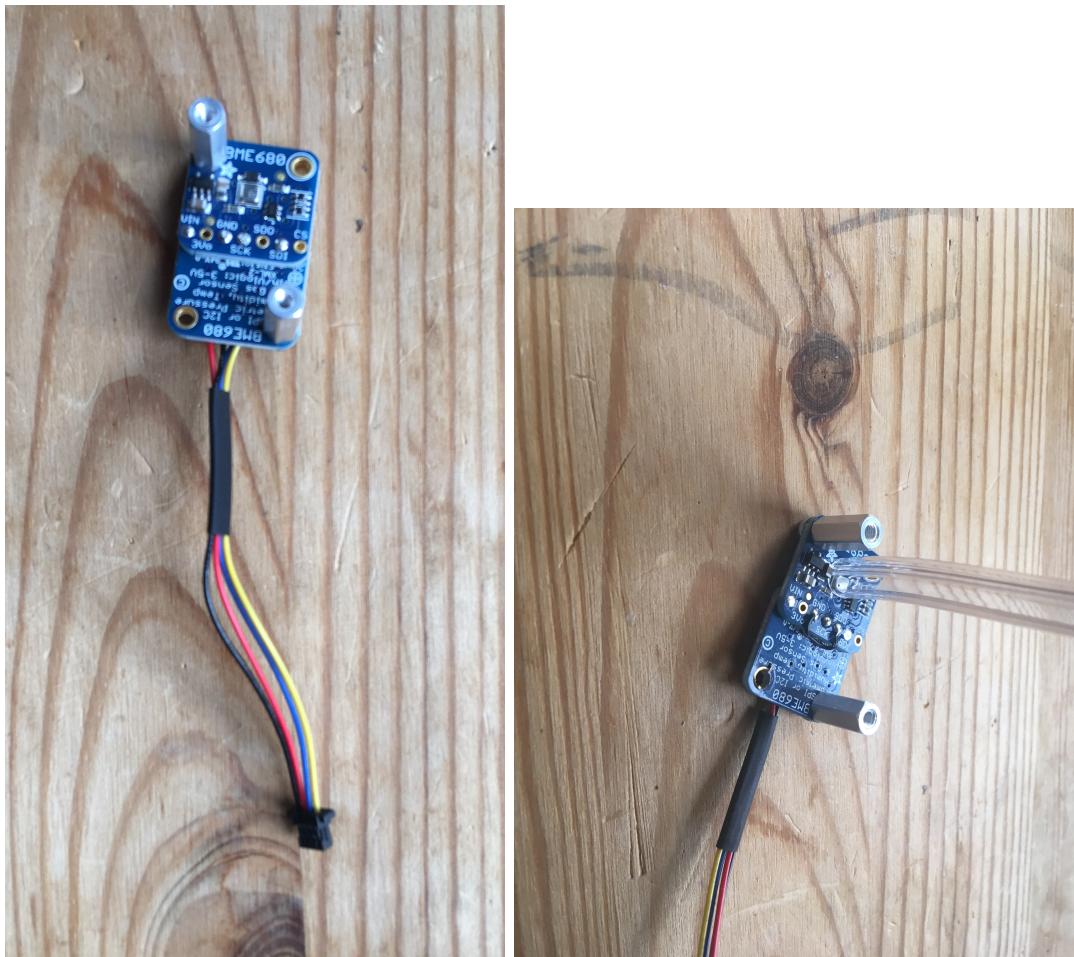


Figure 9: Standoffs installed on pressure sensor assembly and oxygen hose prepared for gluing.

## 5.2 Install pressure sensor assembly. Refer to figure 10 on page 14.

- 5.2.1 Add two standoffs to opposite corners of the pressure sensor assembly
- 5.2.2 These should correspond to the mounting holes you drilled in the enclosure
- 5.2.3 Thread the bare end of the oxygen hose through its hole in the enclosure from the inside until the sensor assembly is pressed against the wall of the enclosure
- 5.2.4 Attach the standoffs into the enclosure from the outside using two machine screws

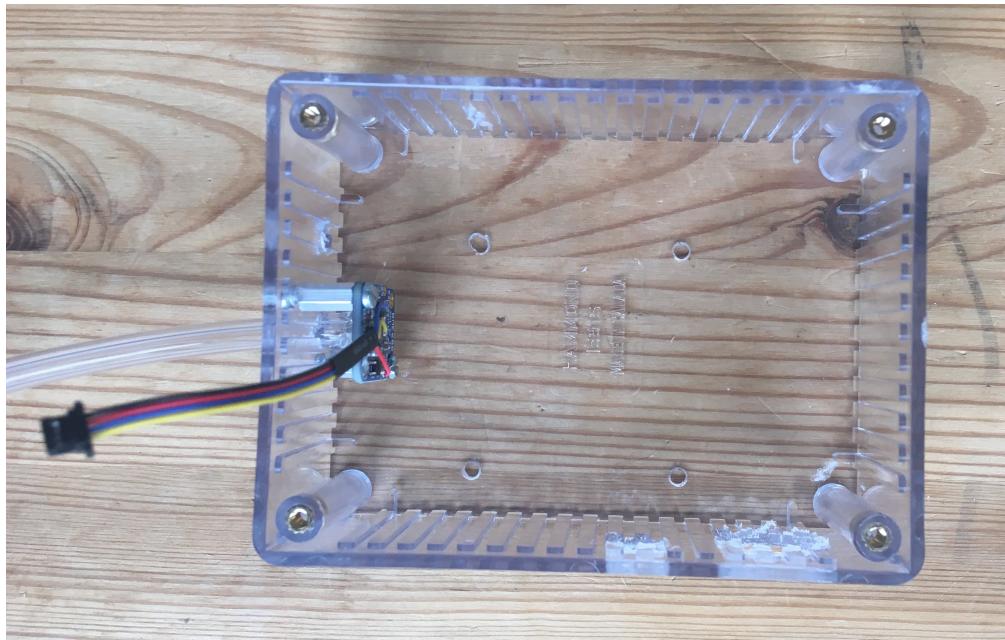


Figure 10: Pressure sensor assembly installed in enclosure.

### 5.3 Zip tie sensor cable to enclosure. Refer to figure 11 on page 14.

- 5.3.1 Thread the small zip tie through the two mounting holes in the enclosure
- 5.3.2 Push the qwiic connector end of the flow sensor cable through its hole in the enclosure. It's a tight fit.
- 5.3.3 Zip tie the cable to the side of the enclosure making sure that the jacket of the USB cable is engaged by the zip tie

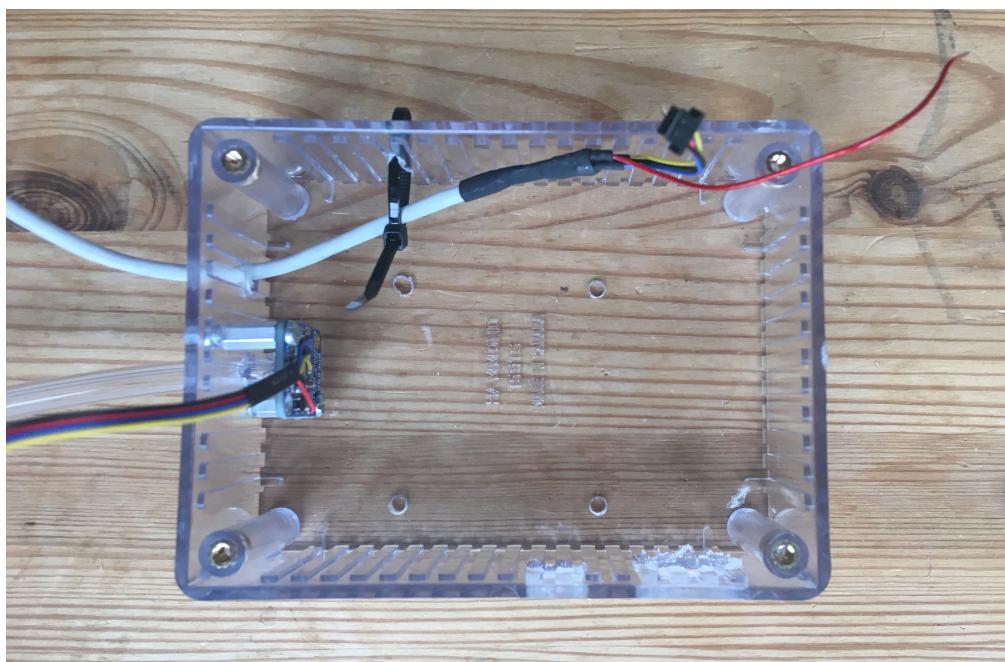


Figure 11: Flow sensor cable with strain relief.

5.4 Install ethernet shield and processor and add final touches. Refer to figure 11 on page 14.

- 5.4.1 Attach standoffs to the four corners of the ethernet shield
- 5.4.2 Screw the standoffs into the enclosure from the bottom
- 5.4.3 Place the ESP32, OLED wing, and Qwiic shield on top of the ethernet shield as depicted in the diagram
- 5.4.4 Plug the flow sensor and pressure sensor assembly into the qwiic shield
- 5.4.5 Add a small piece of electrical tape every 5 inches attaching the oxygen tubing and flow sensor cable
- 5.4.6 Wrap a piece of electrical tape around the flow sensor covering the exposed pads of the flow sensor
- 5.4.7 Add identification card on the next page card to enclosure.

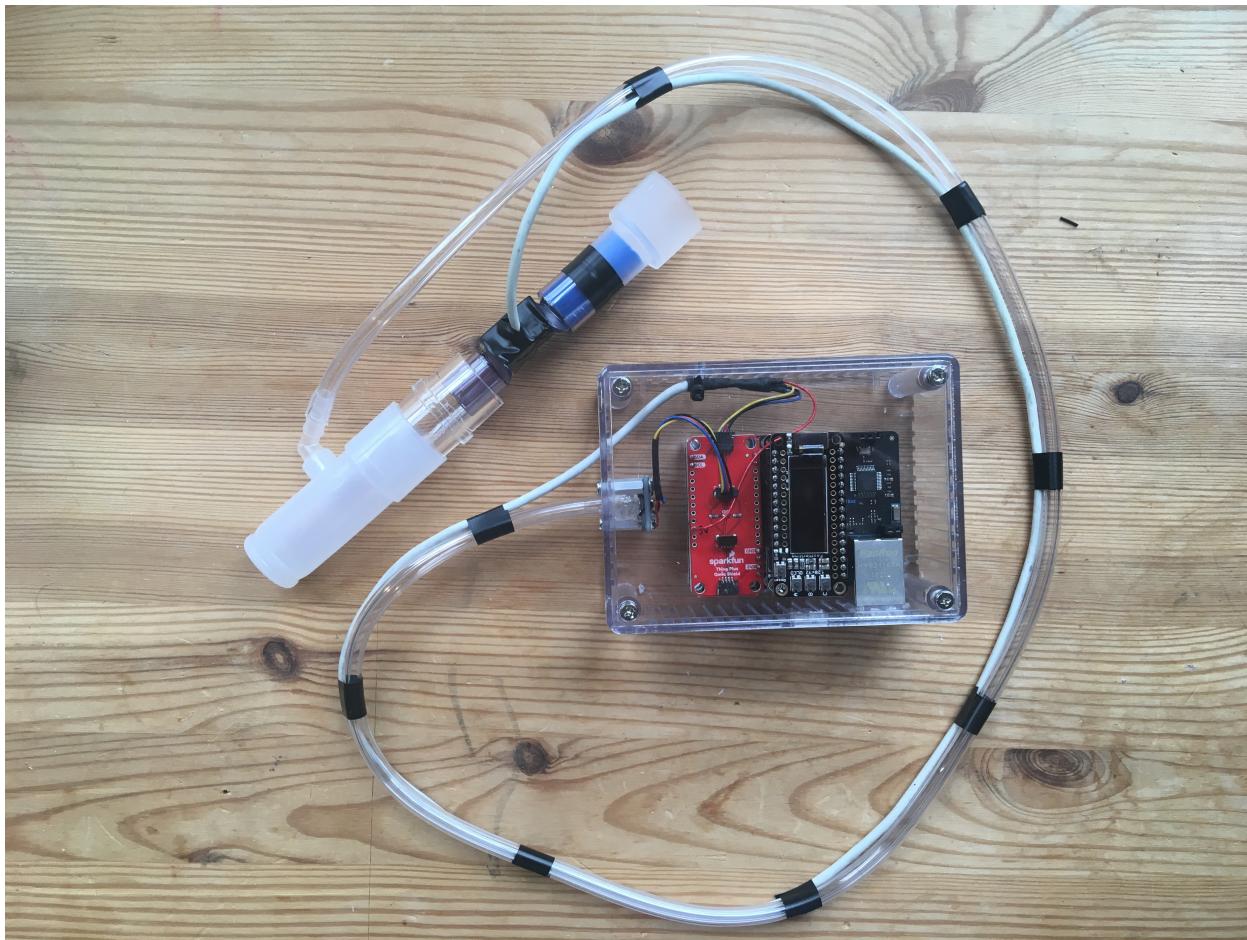


Figure 12: Final assembly.

## 5.2 PCB Based VentMon (T0.3)

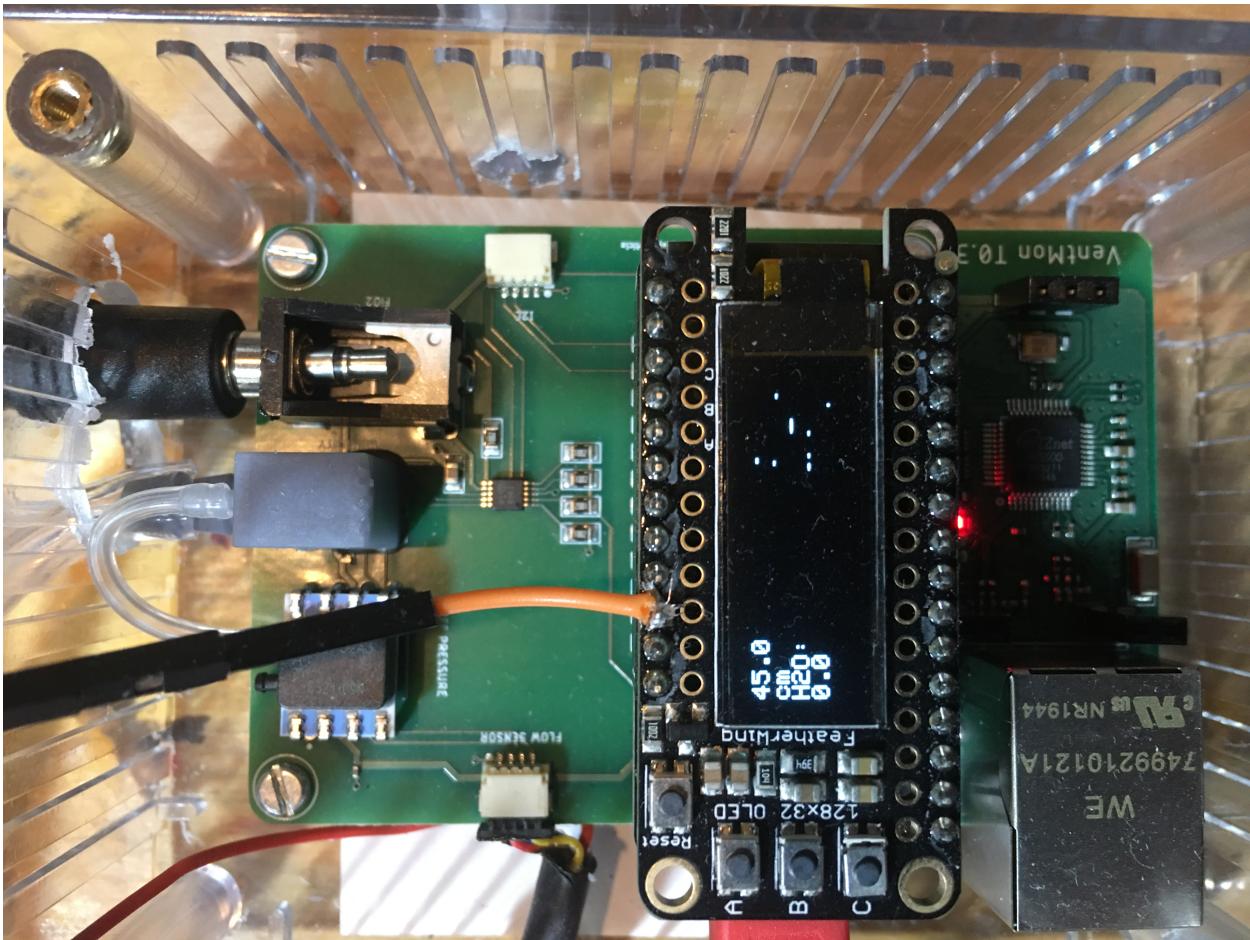


Figure 13: Final assembly of the PCB version

This PCB-based T0.3 version of VentMon (see Figure 13) requires two 3D printed parts – one encapsulated an on-board pressure sensor and one is an airway adaptor – as well as a PCB assembly. Before beginning the assembly process make sure that you have manufactured those parts.

### 1. PCB Assembly

#### 1.1 Add standoffs to PCB

1.1.1 take hardware and put it in the holes

#### 1.2 Mount sensor enclosure for BME280 pressure sensor

1.2.1 Insert plastic part into mounting holes on PCB to check fit and alignment. The barb should face toward the outer edge of the PCB.

1.2.2 Apply epoxy glue to bottom edge and mounting pegs of plastic.

1.2.3 Insert plastic back into holes being careful not to get any glue on the sensor.

1.2.4 Allow 24 hours for epoxy glue to cure before attaching a hose to the barb.

### 2. Enclosure

Follow assembly instructions in section [2](#).

### 3. Flow Sensor Assembly

Follow assembly instructions in section [4](#).

### 4. Oxygen Sensor Assembly

4.1 Screw the airflow flange on to the oxygen sensor

4.2 Plug sensor into the blue airway adaptor piece

4.3 Plug one end of mono audio cable into bottom of oxygen sensor

4.4 Plug the other end into the audio connector on the PCB

### 5. Final Assembly

4.1 Connect Qwiic connector.

4.2 Screw lid on.

4.3 Optionally apply hot-melt glue to Qwiic cable connector where it enters the port.

## 6. Operation instructions

*Provide detailed instructions for the safe and proper operation of the hardware.*

- *Step-by-step operational instructions for operating the hardware.*
- *Use visual instructions as necessary.*
- *Highlight potential safety hazards.*

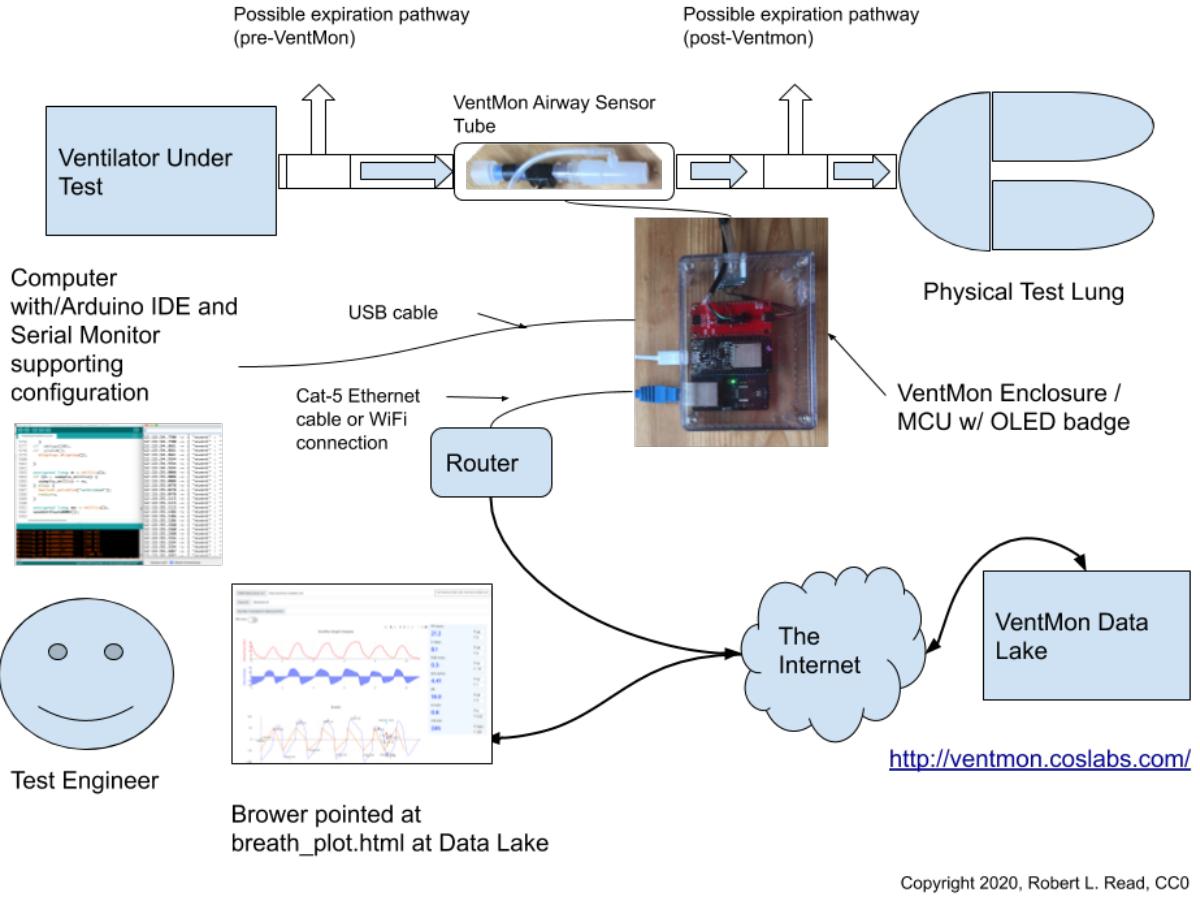


Figure 14: Ventilator Test Usage

### 6.1 VentMon T0.3

*The VentMon T0.3 is not meant to be used on human patients.* The basic usage is to plug the VentMon 22mm airway in between a respiration assistance device being tested (such as a pandemic ventilator or CPAP machine) and a simple physical test lung which models compliance and resistance of the human lung (see Figure 14).

There are four ways to get data from the VentMon of varying quality and convenience with the firmware at the time of this writing.

1. After startup, the VentMon will begin displaying a real-time pressure graph on its small OLED screen, with a scale of 40 cmH<sub>2</sub>O. This display is too small for testing performance, but is a nice check that the breathing circuit is in place and functioning. You can “smoke test” this by breathing into the VentMon and plugging one end with your hand.
2. The VentMon spews a LOT of data out on the serial port in the PIRDS format. Using the serial monitor build into the Arudino IDE, you can visually inspect this data. The volume of the data makes this a cumbersome. You are free to write a serial-port reader of your own for programmatic interpretation.
3. At startup, by default configuration (it can be enabled or disabled), the VentMon tries to see if their is a cable connected to the Ethernet RJ45 jack. Plug the other end of the cable into

your router. If found and enabled, the VentMon will begin sending PIRDS data to the logger at [ventmon.coslabs.com](http://ventmon.coslabs.com). Your data can be viewed in real time by pointing a browser at the URL. Your IP address will appear near the top of the list with other recent or current users; by clicking on “Breath Plot” you will be able to see a complete clinical display.

4. Similarly at startup, the VentMon will attempt to establish a WiFi connection. Using the WiFi connection will transmit exactly the same PIRDS data the logger at [ventmon.coslabs.com](http://ventmon.coslabs.com) and can be used in precisely the same way. However, the VentMon has to know a valid SSID and password to make the WiFi connection. This is configured via the Arduino serial monitor by sending a “c” character (for configure) to the VentMon. It will then pause operation for 10 seconds and give instructions for entering the SSID and password. In this same configuration you can enable or disable the WiFi and the Ethernet connection if you choose. These settings are stored in the EEPROM, and survive a loss of power.

Pressing the “C” button on the VentMon OLED badge sends a special signal to the PIRDS logger to “rotate the log” from this IP address. At the public data lake the current file will be copied to a file which a date and timestamp.

## 6.2 VentMon COTS Version

A practitioner who makes their own version of the VentMon from commercial off-the-shelf parts without ordering a PCB will likely understand how to install and modify the firmware. The VentMon v0.2T[5] version is tagged specifically for this hardware release. The usage will likely be the same, though at the time of this writing some features may not be available in the software they are using, such as WiFi configuration.

## 7. Validation and characterization

The primary use of the VentMon is to develop and verify that rapidly manufactured ventilator meets requirements specification, such as those published by the UK MHRA[2].

Because the VentMon produces pressure and flow graphs at about 25 samples per second which is fast compared to the physiological process of breathing, it is easy to have confidence in the basic dynamic function of the VentMon by simply breathing through it into a plastic test lung. One can easily see that flow and pressure dynamically match the breath or changes in the mechanical ventilation, or even changes induced by changes in the resistance of compliance of the test lung.

The flow and differential pressure sensors used in the VentMon require no calibration. The oxygen sensor is self-calibrated and good enough for accurately measuring FIO<sub>2</sub> mixing in mechanical ventilation. Flow was checked with a graduated syringe and found to be quite accurate.

Other aspects of ventilation such as respiration rate, peak inspiratory pressure, Inhalation to Exhalation (I:E) ratio, work of breathing, etc. are dependent on the VentDisplay software which are a module not technically part of the VentMon hardware. In particular, our software measures pressure rise time which we have argued is an under-appreciated performance characteristic of rapidly manufactured emergency ventilators[9]. This software has been shown to be valuable, although it is dependent on accurate software decisions about the beginning and the ending of a breath, which can be subtle in some situations.

The VentMon published data in the Public Invention Respiration Data Standard format [], which we created to provide a standard for multiple teams to cooperate. A consequence of using this standard and IoT cloud-based software is the ability to support a distributed engineering team,

which has been almost required by the COVID-19 social distancing measures. We aspire to someday support telemedicine as suggested by others[8].

In fact we hope the VentMon will evolve into an open-source component that can be freely integrated into a pandemic ventilator or simply “bolted on” to be used to control ventilators or to monitor human patients. Although it can effectively test ventilation equipment today, before it can be recommended for use on patients, it will require regulatory approval, such as FDA approval. Before that can happen, the VentMon must progress beyond a number of limitations:

1. The current device is not autoclavable or easy to sanitize. Use on a human patient would therefore require it to be disposed of immediately.
2. Our software has not been subject to rigorous Quality Assurance (QA) and Risk Analysis (RA) as the FDA would request. We have argued[7] for an ecosystem in which design teams such as Public Invention produce extensive documentation including QA/RA documentation to be reused by firms that become medical device manufacturers incorporating open-source designs.
3. The connectors on the current design come loose too easily; this is of no importance to a team of test engineers but would be a critical safety problem for use on humans.

Additionally, although the current data lake and software are completely adequate for testing a ventilator and in fact the PIRDS logging system records all data for days or weeks to allow longitudinal study of ventilator reliability, we have not yet written software to allow this data to be analyzed or accessed across long periods of time.

## **8. Acknowledgements**

Robert L. Read conceived of and wrote most of the firmware for the VentMon. Geoff Mulligan created the initial cloud-based IoT approach which became the PIRDS-logger. Lauria Clarke did most of the hardware design, including the assembly instructions and the printed circuit board based on an open source design from Adafruit[] and the SFM3X00 software repo for encapsulating Sensirion flow sensors in the Arduino environment. This paper was written by Robert L. Read and Lauria Clarke.

The Mozilla Open Source Software foundation gave Public Invention a grant that allowed us to develop the hardware and provide 20 VentMons to open source teams around the world free-of-charge, including paying shipping costs. Protocol Labs provided Public Invention a grant that has been used for web communication and conference costs.

We would like to thank early adopters of the VentMon on pandemic ventilator teams such as the ARMEE[3] and DIY-Beatmungsgerät.de[10] who gave us valuable bug reports and feedback.

Adafruit graciously continued to supply pandemic engineers and researchers even when their normal business was disrupted.

Student volunteers on the COVID19-vent-list project[6] helped to make the need for the VentMon clear by assessing over 100 pandemic ventilator teams.

## **9. Declaration of interest**

None.

## References

- *Include at least one reference, to the original publication of the hardware you customized.*
- *Include other references as required. Include references to put your device in context in the literature. For more information on the reference format in HardwareX please see the Guide for Authors at: <https://www.elsevier.com/journals/hardwarex/2468-0672/guide-for-authors>*

## References

- [1] Lauria Clarke. *VentMon v0.3T PCB*. Version v0.3T. Aug. 2020. URL: <https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/tree/master/design/pcb>.
- [2] UK MHRA. *Specification for ventilators to be used in UK hospitals during the coronavirus (COVID-19) outbreak*. 2020.
- [3] *Millions of Ventilators*. Oct. 2020. URL: <https://armeevent.com/>.
- [4] Robert L. Read. *VentDisplay v0.3T Release*. Version v0.3T. Aug. 2020. doi: [10.5281/zenodo.4079374](https://doi.org/10.5281/zenodo.4079374). URL: <http://doi.org/10.5281/zenodo.4079355>.
- [5] Robert L. Read. *VentMon T0.2 Firmware Release*. Version T0.2. Aug. 2020. URL: <https://github.com/PubInv/ventmon-ventilator-inline-test-monitor/releases/tag/v0.2>.
- [6] Robert L. Read et al. *COVID-19 Vent List*. Oct. 2020. URL: [https://docs.google.com/spreadsheets/d/1inYw5H4RiL0AC\\_J9vPWzJxXCdlkMLPBRdPgEVKF8DZw/edit#gid=0](https://docs.google.com/spreadsheets/d/1inYw5H4RiL0AC_J9vPWzJxXCdlkMLPBRdPgEVKF8DZw/edit#gid=0).
- [7] Robert L. Read et al. *The Pandemic-inspired Case for an Open-Source Medical Hardware Ecosystem VentDisplay v0.3T Release*. Sept. 2020. URL: <https://makezine.com/2020/09/21/the-pandemic-inspired-case-for-an-open-source-medical-hardware-ecosystem/>.
- [8] Gregory B Rehm et al. “Development of a research-oriented system for collecting mechanical ventilator waveform data”. In: *Journal of the American Medical Informatics Association* 25.3 (2018), pp. 295–299.
- [9] EB Schulz and RL Read. “The importance of characterising dynamic response and inertia in potential rapidly manufactured ventilator systems”. In: *Anaesthesia* (July 2020). doi: [10.1111/anae.15190](https://doi.org/10.1111/anae.15190).
- [10] *WILLKOMMEN BEIM PROJEKT DIY-BEATMUNGSGERÄT*. Oct. 2020. URL: <https://www.xn--diy-beatmungsgert-5qb.de/>.