HardwareX 12 (2022) e00358

Contents lists available at ScienceDirect

HardwareX

journal homepage: www.elsevier.com/locate/ohx

CAREDAQ: Data acquisition device for mechanical ventilation waveform monitoring

Qing Arn Ng [a], Christopher Yew Shuen Ang [a,*], Yeong Shiong Chiew [a,*], Xin Wang [a], Chee Pin Tan [a], Mohd Basri Mat Nor [b], Nor Salwa Damanhuri [c], J. Geoffrey Chase [d]

[a] School of Engineering, Monash University Malaysia, Subang Jaya, Selangor 47500, Malaysia
[b] Kulliyah of Medicine, International Islamic University Malaysia, Kuantan, Pahang 25200, Malaysia
[c] Faculty of Electrical Engineering, Universiti Teknologi MARA, Cawangan Pulau Pinang, 13500, Permatang Pauh, Pulau Pinang, Malaysia
[d] Center of Bioengineering, University of Canterbury, Christchurch 8041, New Zealand

ARTICLE INFO

Article history:
Received 25 June 2022
Received in revised form 1 September 2022
Accepted 5 September 2022

Keywords:
Mechanical ventilation
Ventilator waveform data
Data acquisition device

ABSTRACT

Mechanical ventilation (MV) provides respiratory support for critically ill patients in the intensive care unit (ICU). Waveform data output by the ventilator provides valuable physiological and diagnostic information. However, existing systems do not provide full access to this information nor allow for real-time, non-invasive data collection. Therefore, large amounts of data are lost and analysis is limited to short samples of breathing cycles. This study presents a data acquisition device for acquiring and monitoring patient ventilation waveform data. Acquired data can be exported to other systems, allowing users to further analyse data and develop further clinically useful parameters. These parameters, together with other ventilatory information, can help personalise and guide MV treatment. The device is designed to be easily replicable, low-cost, and scalable according to the number of patient beds. Validation was carried out by assessing system performance and stability over prolonged periods of 7 days of continuous use. The device provides a platform for future integration of machine-learning or model-based modules, potentially allowing real-time, proactive, patient-specific MV guidance and decision support to improve the quality and productivity of care and outcomes.

Specifications table

Hardware name	CAREDAQ (***C****linical* ***A****pplication of* ***R****espiratory* ***E****lastance trial* ***D****ata* ***Acq****uisition device)*
Subject area	• Mechanical Engineering • Biomedical Engineering • Medical sensor
Hardware type	• Mechanical engineering and materials science • Clinical tool for ventilator • Data acquisition device

(*continued on next page*)

* Corresponding authors.
E-mail addresses: Christopher.Ang@monash.edu (C. Yew Shuen Ang), chiew.yeong.shiong@monash.edu (Y. Shiong Chiew).

https://doi.org/10.1016/j.ohx.2022.e00358
2468-0672/

(continued)

Hardware name	CAREDAQ (***C**linical **A**pplication of **R**espiratory **E**lastance trial **D**ata **Acq**uisition device)*
Closest commercial analog	*No commercial analog is available.*
Open-source license	*Creative Commons Attribution-ShareAlike 4.0*
Cost of hardware	*~ $ 250 USD*
Source file repository	*https://data.mendeley.com/datasets/vnvn9m58p2/draft?a=3a509441-0e7b-4699-bdfc-c2d103e77f92*

1. Hardware in context

Mechanical ventilation (MV) patients need to be continuously monitored to ensure safe ventilation [1]. Modern ventilators display real-time airway pressure and flow waveforms delivered to the patient, enabling real-time evaluation, which is vital for assessing and ensuring safe ventilation settings based on patient-specific conditions [2]. However, these waveform data are not stored, and both clinical studies and regular care rely on clinicians for bedside diagnosis and management of MV [3–5]. Hence, there is a need for an automated data acquisition system to acquire data automatically at the bedside for real-time data monitoring and analysis, storage, and further retrospective analysis.

Internet of things (IoT) devices, such as Raspberry Pi (RPi), have gained popularity in data acquisition and monitoring applications [6–9]. IoT-based systems, such as PIMAP, can autonomously monitor patients in real-time [10]. Similarly, Rehm *et al.* [11] also developed a research-oriented system for collecting ventilator waveform data using RPi microcomputers and software frameworks to allow automated data collection. Importantly, this study validates the generalizability and utility of a continuous, multi-patient data collection system of ventilator waveform data (VWD) in the healthcare system. However, it does not provide data analysis to the user during operation and its design lacks a proper mounting solution for scalable bedside monitoring deployment.

This paper describes the process of developing an automated, continuous, and non-invasive data acquisition device (DAQ) for MV waveform monitoring. Specifically, the DAQ system is named CAREDAQ (**C**linical **A**pplication of **R**espiratory **E**lastance trial **D**ata **Acq**uisition). The primary function is to collect airway pressure and flow data output from a mechanical ventilator and enable retrospective monitoring and analysis of the data.

In particular, this data can be further processed to provide respiratory mechanics and other ventilatory information not available on today's ventilators, but useful to personalise and guide MV treatment [12–16]. CAREDAQ thus provides a platform for future development and integration of software modules, including machine learning models [17–19] or model-based algorithms [14,16,20,21] which could potentially provide real-time, proactive, and patient-specific MV decision support and care, improving care and outcomes.

2. Hardware description

The CAREDAQ device presented in this work consists of a portable stand-alone DAQ device based on RPi to acquire data from a mechanical ventilator. The ventilator used in this study is the Puritan Bennett 980 (PB-980) ventilator (Medtronic, Dublin, Ireland). During MV treatment, the ventilator provides valuable information to caregivers via the user interface of the PB-980 ventilator, such as airway pressure and flow data (Fig. 1). By using the waveform output function of the PB-980 ventilator, airway pressure and flow data can be exported for analysis via the ventilator's RS-232 serial port to be recorded by the DAQ.

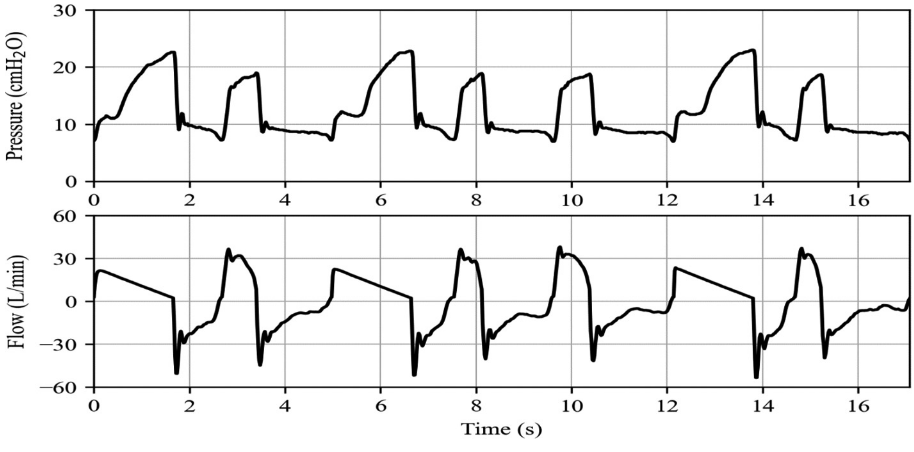

Fig. 1. Raw VWD plot. The top plot displays the airway pressure waveforms, while the bottom plot shows the airway flow waveforms.

The computer-aided design (CAD) of the DAQ device shown in Fig. 2 consists of the main device body and mounting parts for mounting on the ventilator railing. The electronics hardware is contained in a custom 3D printed casing shown in Fig. 3. The assembled DAQ has overall dimensions measuring 196.3 mm × 105.9 mm × 167.95 mm (length × width × height). The DAQ connects to the ventilator via an RS-232 cable and reads the ventilator data stream. The main processing unit, based on an RPi 4B (4 GB) single-board computer (SBC), is responsible for acquiring the ventilator waveform data, processing the data, and providing user interaction. Additional parts in the device include a touchscreen, fan, heatsink, and real-time clock (RTC) module. The touchscreen display is used to display data and interact with the user. The fan and heatsink are used to cool the SBC. The RTC module is used to provide accurate timing when the device is powered off or not connected to the network. The wiring connections of the additional components to the Raspberry Pi are shown in a schematic diagram in Fig. 4.

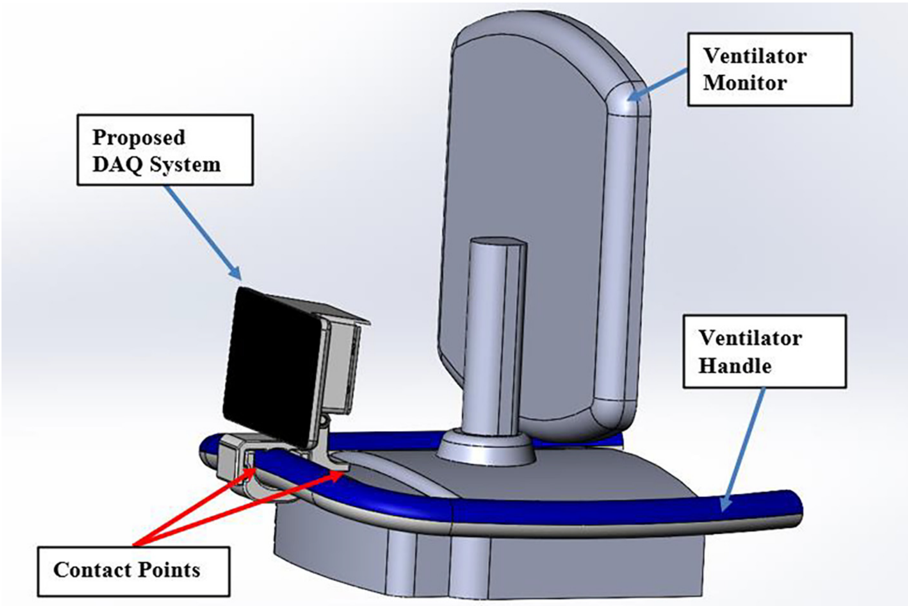

Fig. 2. 3D CAD concept model of the proposed DAQ system attached to the back of the Puritan Bennett PB-980 mechanical ventilator. The top clamp rests on the handle railing and body of the ventilator. The DAQ device can swivel around the top clamp to provide the best viewing angle. Foam pads are added along contact points (marked in red arrows) between the ventilator and clamping mechanism to provide extra grip and padding. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

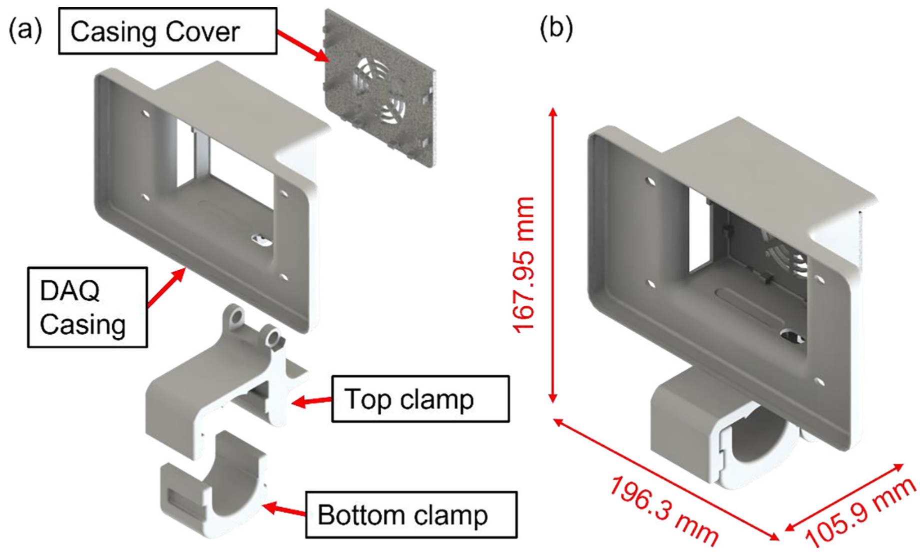

Fig. 3. (a) The 3D printed components of the DAQ consist of the Casing Cover, DAQ Casing, top clamp and bottom clamp. (b) The overall dimensions of the assembled DAQ.

In addition, a data output simulator has been developed to simulate the data output of the PB-980 ventilator. The simulator features an Arduino Uno board (Arduino AG, USA) as data output, and the board is connected to the DAQ's USB port via a USB type-A to USB type-B cable. Once connected, the script onboard outputs breathing cycles in a continuous loop with

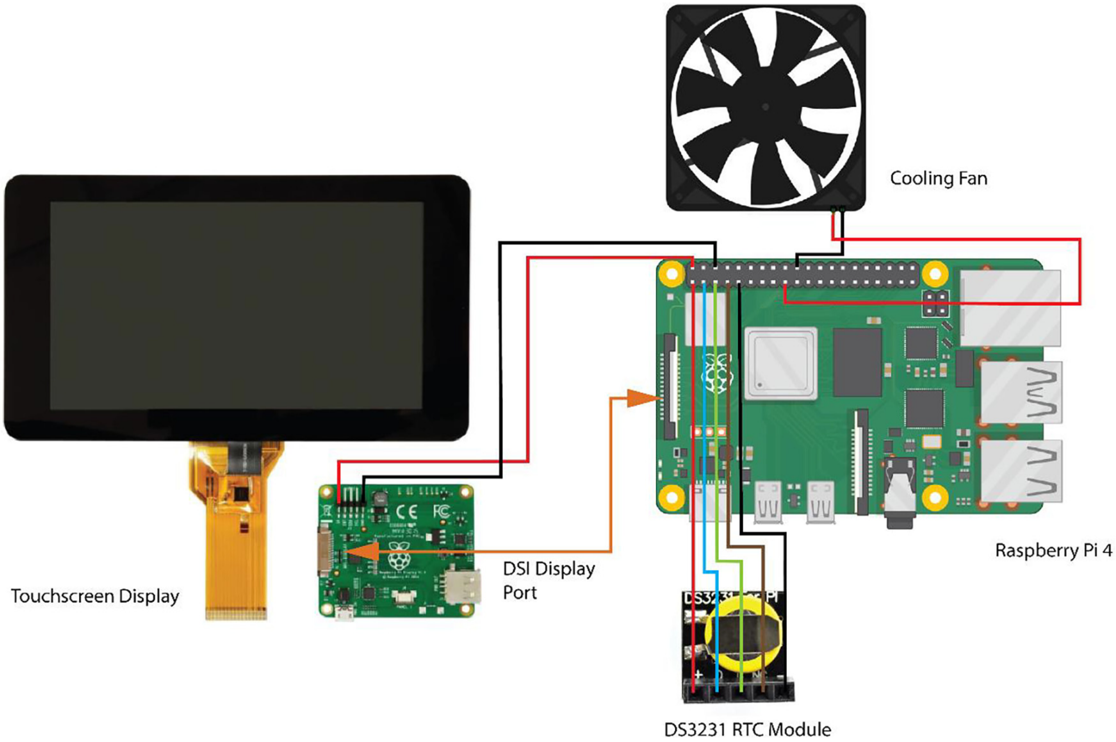

Fig. 4. Schematics diagram of wiring connection to the Raspberry Pi.

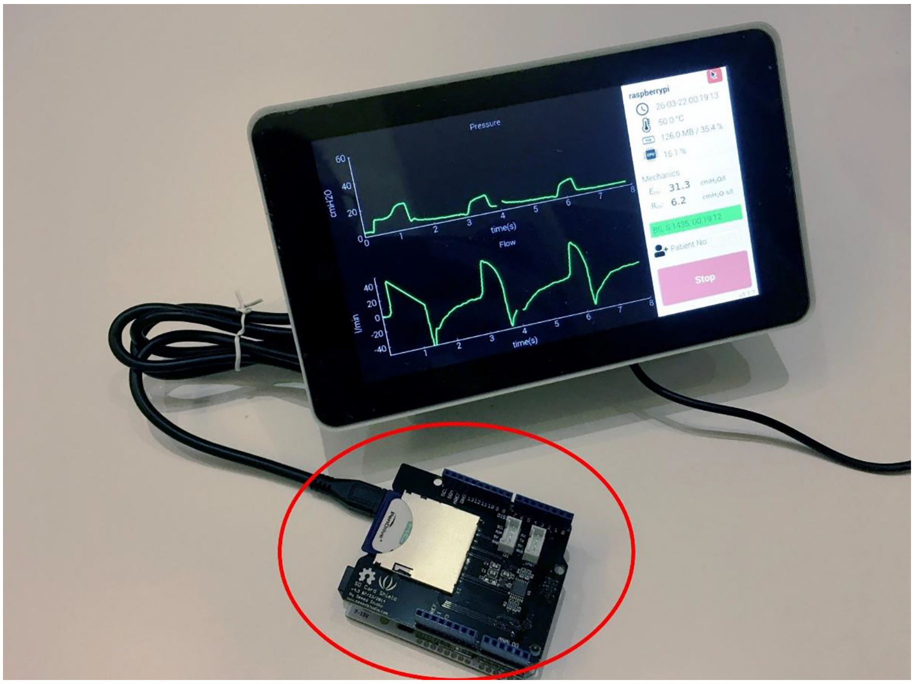

Fig. 5. Figure showing DAQ recording data from a data output simulator (Red circle) for system testing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

retrospective VWD loaded on the SD card's memory. The data output simulator enables system testing of the continuous recording ability of multiple DAQs without connecting them to ventilators (Fig. 5), which clinically, can be in short supply due to regular use in a given intensive care unit.

3. Design files

The design files include three-dimensional drawings of all components as STL and SOLIDWORKS files, as well as the code for the data output simulator as outlined and described below:

3D printed parts

The DAQ casing, casing cover, top clamp, and bottom clamp were all 3D printed using Flashforge (Zhejiang Flashforge 3D Technology Co., ltd, China) ABS white filament. Our prototype was produced using Flashforge Inventor 3D printer. We recommend a printing temperature of at least 215 °C, and a bed temperature of at least 100 °C, in an enclosed printing chamber to prevent warping. The 3D-printed casing can be sterilized after use, meaning the device can be reused and redeployed to other ventilators for data acquisition.

3D modelled parts

The modelled parts were used mainly for visualization purposes, using a 3D computer-aided design assembly.

Software

The data acquisition software application comprises a three-layer architecture: Data layer, Application layer and Presentation layer. The layers are isolated from each other to allow the implementation of different layer-specific functions and interfaces. Hence, the architecture allows added versatility to the customisation of software where new modules can be added for use and adaptation to different ventilator models. Specifically, the data layer can be extended so that different ventilators with different outputs can be added and processed to the same data structure for standardisation and comparison. This data is then processed in the Application layer before being passed to the Presentation layer. The graphical user interface (GUI) can be modified to include other ventilator models as seen in Fig. 11.

The DAQ's graphical user interface (GUI) is controlled by Python (3.6, Python Software Foundation, Wilmington, DE, USA) software. The GUI was created using the *PyQt5* library. The main file for the code of DAQ and install instructions are included as design files.

Design files summary

Design file name	File type	Open-source license	Location of the file*
bottom_clamp.sldprt	SLDPRT	CC BY-SA 4.0	Folder: 'Hardware'
casing_cover.sldprt	SLDPRT	CC BY-SA 4.0	Folder: 'Hardware'
daq_casing.sldprt	SLDPRT	CC BY-SA 4.0	Folder: 'Hardware'
top_clamp.sldprt	SLDPRT	CC BY-SA 4.0	Folder: 'Hardware'
bottom_clamp.stl	3D-printable	CC BY-SA 4.0	Folder: '3D files'
casing_cover.stl	3D-printable	CC BY-SA 4.0	Folder: '3D files'
daq_casing.stl	3D-printable	CC BY-SA 4.0	Folder: '3D files'
top_clamp.stl	3D-printable	CC BY-SA 4.0	Folder: '3D files'
CARESoft User Guide_Hardware X	Microsoft Word Document	CC BY-SA 4.0	Folder: 'User Guide'
DAQDOS.ino	Arduino	CC BY-SA 4.0	Folder: 'Software'
CAREDAQ_installation_video_final.mp4	Video	CC BY-SA 4.0	Folder: 'Video'

* The location of design files is relative to the root directory of source files: https://data.mendeley.com/datasets/vnvn9m58p2/draft?a=3a509441-0e7b-4699-bdfc-c2d103e77f92.

4. Bill of materials

Designator	Component	Qty	Unit cost (USD)	Total cost (USD)	Source of materials	Material type
Hardware:						
Raspberry Pi 4	Raspberry Pi 4 Model B – 4 GB RAM	1	55.00	55.00	https://www.adafruit.com/product/4296	Electronics
Touchscreen display	Official Raspberry Pi touchscreen display	1	79.95	79.95	https://www.adafruit.com/product/2718	Electronics
Power supply	Official Raspberry Pi power supply 5.1 V 3A with USB C	1	7.95	7.95	https://www.adafruit.com/product/4298	Electronics
Surge Protector	Belkin 1-Outlet Surge Protector	1	8.84	8.84	https://amzn.to/3ieCcXK	Electronics
RTC module	DS3231 RTC module for Raspberry Pi	1	3.33	3.33	https://my.cytron.io/p-ds3231-rtc-module-for-raspberry-pi	Electronics
Cooling Fan	DC 5 V 5010 Cooling Fan 50x10mm, 2Pin	1	1.31	1.31	https://bit.ly/3ihizON	Electronics
microSD card	Samsung EVO Plus microSD 128 GB	1	19.10	19.10	https://amzn.to/3CRhYwB	Electronics
Heatsink	Raspberry Pi 4 Heatsink	1	1.19	1.19	https://bit.ly/3JoRmpf	Metal
M3 × 20 mm bolt	M3 × 16 mm bolt	4	0.02	0.08	https://my.cytron.io/c-bolts/p-bolt-m3x20mm	Metal
M3 × 6 mm bolt	M3 × 6 mm bolt	4	0.01	0.04	https://my.cytron.io/c-bolts/p-bolt-m3x6mm	Metal
M8 × 45 mm hex socket cap screw	Stainless Steel Hex Socket Cap Screw, SUS 304 DIN 912 M8 × 45 mm	1	2.41	2.41	https://my.rs-online.com/web/p/socket-screws/7976272	Metal
M8 hex nut	Stainless Steel, Hex Nut, M8	1	2.45	2.45	https://my.rs-online.com/web/p/hex-nuts/0530769	Metal
USB to Serial cable	UGREEN USB 2.0 to RS232 DB9 Serial Cable	1	6.34	6.34	https://bit.ly/3IzQZHf	Electronics
Data Output Simulator:						
Cytron Uno board	Cytron UNO – Arduino UNO Compatible	1	14.22	14.22	https://bit.ly/3K5E1CM	Electronics
USB Micro B cable	USB Micro B cable	1	0.95	0.95	https://bit.ly/3J4rVZd	Electronics
SD card shield	SD Card Shield V4	1	12.33	12.33	https://bit.ly/3u6ZWnz	Electronics
SD card	SanDisk 32 GB Ultra SDHC UHS-I Memory Card (Any size)	1	9.29	9.29	https://amzn.to/3K802kk	Electronics
3D-printed parts:				39.20	Mass (g):	
DAQ Casing		1			85.66	Plastic
Casing Cover		1			16.42	Plastic
Top Clamp		1			40.03	Plastic
Bottom Clamp		1			22.63	Plastic
Total mass:					164.74*	
Total:				263.98		

* Estimated cost of 3D printing at USD 0.24/gram.

5. Build instructions

3D printed casing

The main casing, casing cover, top clamp, and bottom clamp are 3D printed with an infill of 30 % and a layer height of 0.3 mm. We recommend using ABS filament with a diameter of 1.75 mm. Fig. 6 shows the result of the 3D printed casing attached to components. The casing and mounting design can be modified for different accessories or use cases.

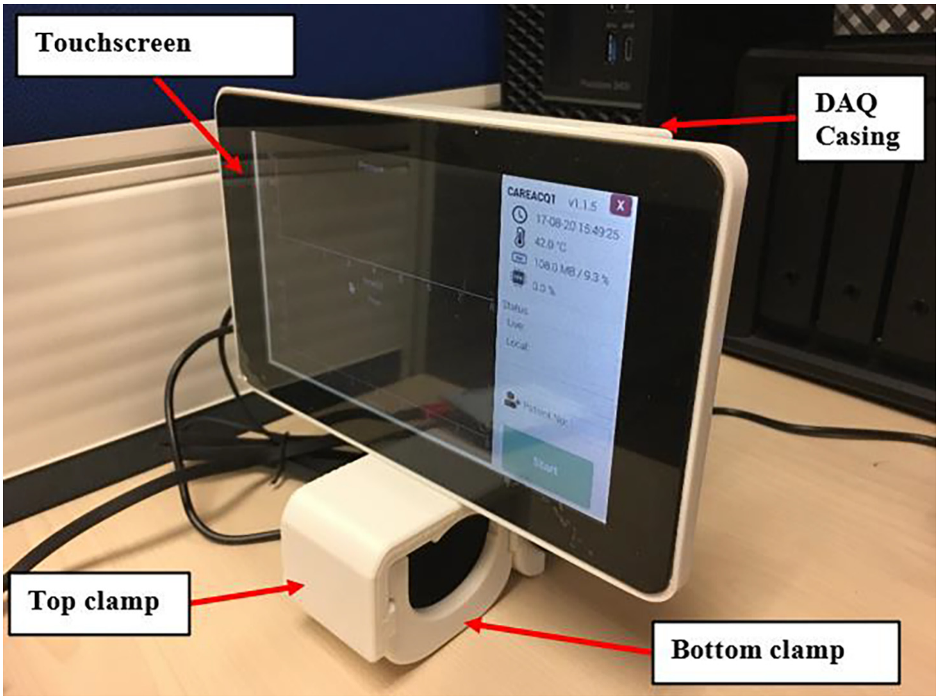

Fig. 6. Figure showing prototype of the DAQ with internal components attached to the casing and mounting parts.

Connect additional modules to Raspberry Pi

First, secure the touchscreen display to the Raspberry Pi (RPi) using four M2.5 × 4.5 mm bolts. Second, flash the microSD card with *Raspberry Pi OS* (buster) and insert the microSD card into the RPi microSD slot. Connect the DSI display cable to the DSI Display port on the RPi and touchscreen display, respectively. The next step was to connect the display, RTC module, and cooling fan with the RPi General-purpose input-output (GPIO) pins, as shown in Fig. 4. The touchscreen display's *5 V pin* is connected to Pin 2/Pin 4 (5 V) of the RPi, and *Gnd* Pin is connected to Pin 6 (Ground) of the RPi. RTC module is slotted into Pin 1,3,5,7,9 of RPi directly. Next, the DAQ casing is secured to the touchscreen display using four M3 × 6 mm bolts. The cooling fan is attached to the Casing cover with four M3 × 16 mm bolts and nuts. The cooling fan's power and ground connection is connected to Pin 17 and Pin 20, respectively to power the fan with a 3.3 V power rail. The Casing cover is attached to the DAQ casing using the clips on the cover. The DAQ top mounting (Top clamp) attaches to the DAQ casing with an M8 hex bolt and an M8 nut. The DAQ bottom mounting slides in the top mounting to complete the assembly process. Fig. 7 shows the complete assembled components of the DAQ. A video of the complete assembly process is shown in the source files.

Attaching the DAQ to the ventilator

Align the Top Clamp of the DAQ to the back of the ventilator and rest it on the railing of the ventilator (Fig. 8). Once in place, slide the Bottom Clamp into the Top Clamp to lock it in place. Next, connect the RPi's USB port to the ventilator's serial port using a USB to serial cable. The cable used in this study is a USB 2.0 to RS232 DB9 Serial Cable with a Prolific PL2303 chipset.

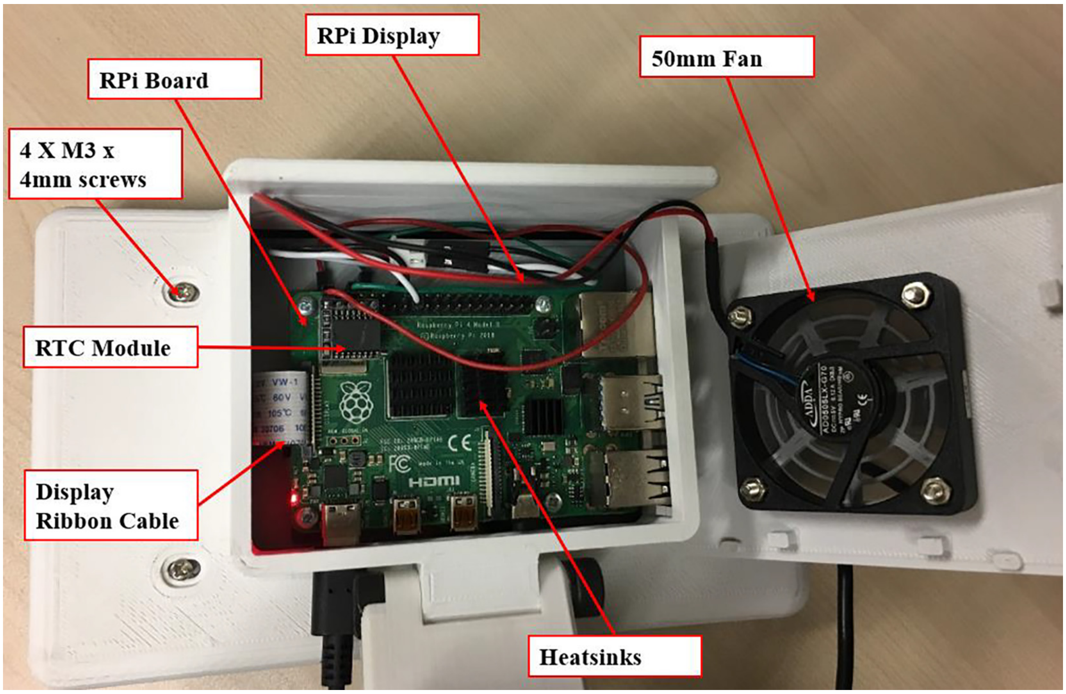

Fig. 7. Inside view of the device showing the internal components layout of the DAQ. The RPi board is attached behind the RPi touchscreen display. Additional modules such as RTC module and fans are connected to the RPi board.

Fig. 8. Figure of DAQ assembly mounted on the railing of the ventilator. The USB to Serial cable is connected to the Raspberry Pi and ventilator, respectively (Red arrows). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6. Operation instructions

Configuring the ventilator

Before starting the data acquisition, the ventilator needs to be correctly set up to output waveform data. Fig. 9 shows the main interface of the ventilator screen. The waveform output settings can be accessed in the *Settings* interface (Fig. 10). Once configured, the ventilator will output airway pressure and flow data once ventilation starts.

Operating the DAQ

First, the DAQ should be powered through the RPi power supply. The program will launch automatically, and the start-up screen will be shown (Fig. 11). As shown in Fig. 11, the DAQ software currently allows operation in two modes as follows:

1) *Puritan Bennett 980 mode*: This mode collects data from the PB980 ventilator in real-time and stores the data in text file format. The parameters are adjusted according to the ventilator's output settings.
2) *Retrospective mode*: This mode allows the simulation of data from retrospective data of the patient. Virtual patients can be simulated in this mode for research purposes. It can enable user training or testing purposes.

Fig. 9. Main ventilator screen of the ventilator interface. *Settings* page is accessed via clicking on the *Settings* button (Red Circle). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

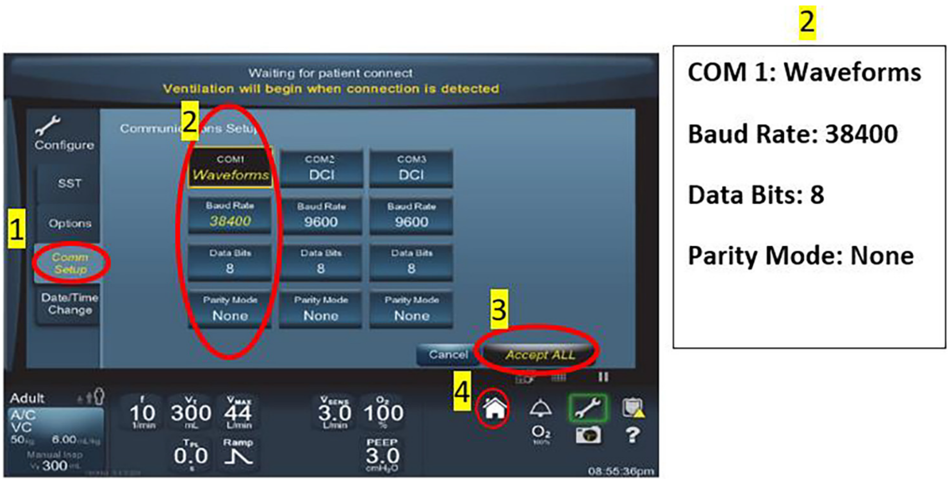

Fig. 10. Settings interface of the ventilator. Waveform output is enabled by the following step: *Comm Setup*[1] -> set accordingly[2] -> press *Accept All*[3]-> press *Home*[4].

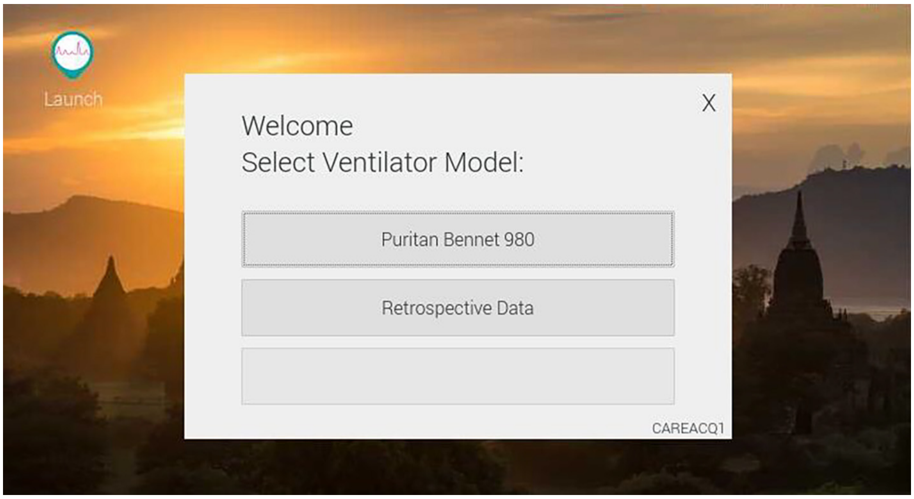

Fig. 11. Start-up screen of DAQ software showing a selection of multiple modes/ventilator types before starting data acquisition software, enabling expansion of modules in the future. (1): Puritan Bennett 980 mode. (2): Retrospective mode. (3): Empty reserved module slot for future expansions.

Fig. 12 shows the GUI of DAQ software in retrospective mode. The GUI of both operation modes is similar with the exception of a label showing *Retrospective mode* to distinguish between the two modes. Table 1 shows the operating procedure of the DAQ software. The recorded data is saved locally in the Raspberry Pi and can be retrieved by accessing the file system. The data can then be exported via flash drive or other means. Step-by-step instruction for operating the CAREDAQ is provided in the attachment.

Fig. 12. GUI of DAQ software is categorized into several main sections. The three main sections are summarized below in a clockwise direction: (1). **Status:** Shows device critical information. (2). **Controls:** Control buttons to set patient numbers and start/stop data acquisition. (3). **Display:** Real-time ventilator output waveform display.

Table 1
DAQ application operating procedure.

Puritan Bennett PB980 mode	Retrospective mode
1. Connect DAQ to a power source to power on the device. 2. Select DAQ operation mode (Puritan Bennett PB980 mode) 3. Input patient number using on-screen keypad. 4. Click ‘Start’ to start collecting data.	1. Connect DAQ to a power source to power on the device. 2. Select DAQ operation mode (Retrospective mode) 3. Input patient number using on-screen keypad. 4. Click ‘Start’ to start collecting data.

7. Validation and characterization

To test the device's operation, multiple tests have been conducted. To demonstrate CAREDAQ's ability to record data continuously, the performance of the system was monitored during a 7-days data recording session. Particularly, the central processing unit (CPU) usage, random access memory (RAM) usage, and disk usage were monitored.

System performance and stability test

To demonstrate the CAREDAQ's ability to record data continuously, we monitor the CAREDAQ when recording data from a continuous data stream of VWD. Key performance indicators monitored in this text include CPU usage, RAM usage, and disk usage. The memory utilisation test is important to ensure that there is no memory leak during prolonged use and recording

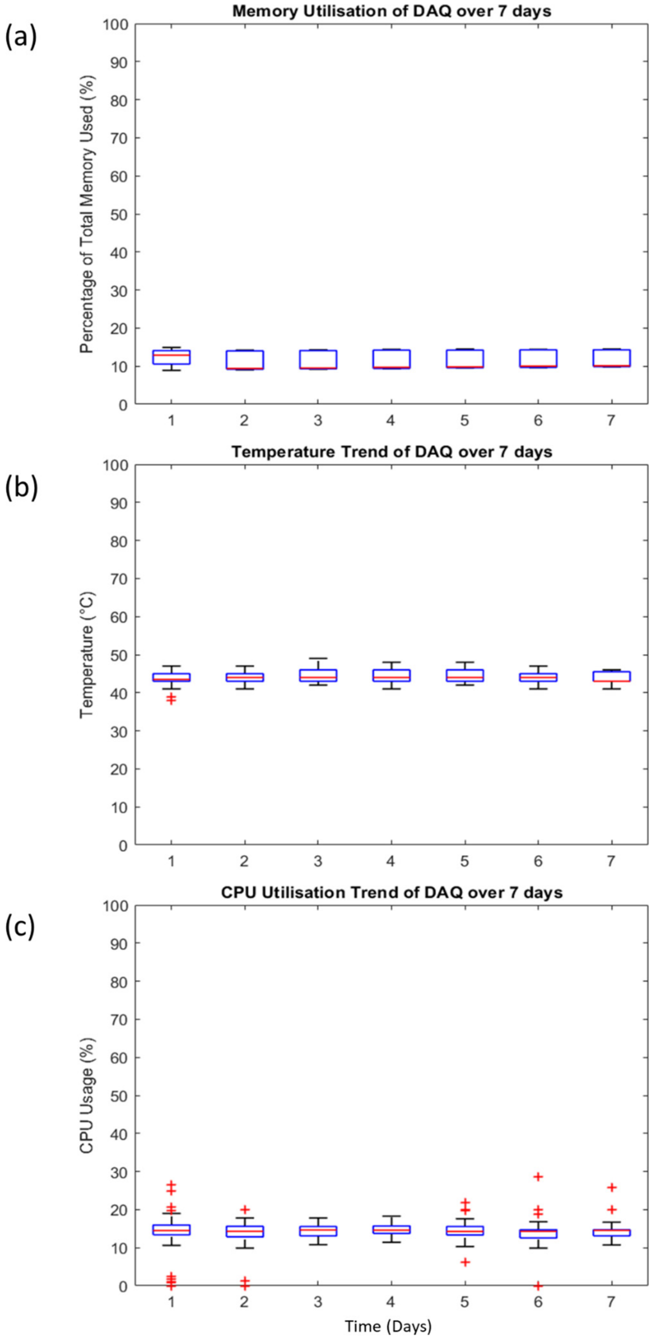

Fig. 13. Box plot showing (a) Memory utilisation; (b) CPU temperature; (c) CPU utilisation trend of DAQ over 7 days.

of data. Besides that, the temperature test ensures that there is sufficient airflow provided to the device to maintain the CPU's operating temperature. The experiment was set up in a room with room conditions of 26-degree Celsius room temperature, and 64 % relative humidity, which is extreme for typical intensive care units under strict climate control. The ventilator is set to ventilate under normal ventilation settings and recorded by DAQ for a 7-day duration.

The memory utilisation usage trend in Fig. 13(a) and CPU temperature trend in Fig. 13(b) show low and stable overall values of mean (±Standard Deviation (SD)) 11.3 % (±SD: 2.20) and 44.15 °C (±SD: 1.70) respectively. The memory test shows that there is no increase in memory utilisation of DAQ from Day 1 (Mean: 13.4 %) – Day 7 (Mean: 11.2 %), and it is well below the limit of 100 % memory utilisation. The RPi's CPU temperature is well within the CPU design operating temperature (0 – 50°Celsius ambient). Thus, the heatsink and fan are sufficient for the cooling of DAQ.

Fig. 13(c) shows the CPU utilisation of DAQ over 7 days. The mean CPU usage is 14.18 % (±SD: 2.86). Overall, the utilisation is low and well within the design limit of the CPU (less than 100 %). Inconsistency and higher data variation recorded in the CPU usage may be related to too low of a sampling rate, which is currently once per hour. Increasing the rate of data sampling may improve the reading of the data. The measurement of the overall system CPU's utilisation may also be affected by system tasks, background processes, or auto backup scripts running in the background, hence the spike in usage is observed.

Conclusion

In summary, this paper presents the CAREDAQ VWD acquisition system design process, including all hardware and software. The data acquisition hardware system requirements and specifications are shown. The hardware and software architecture are detailed, as well as the design process of DAQ, and mounting design. The CAREDAQ could be readily networked to a server, allowing real-time data transmission and potentially live monitoring and analysis of ventilator waveform data.

Ethics statements.

The study was approved by the research ethics committee of the International Islamic University Malaysia with a trial number (IREC 2020-100).

Declaration of Competing 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.

Acknowledgments

The authors would like to thank the Ministry of Energy, Science, Technology, Environment and Climate Change (MESTECC) research grant (IF0219I1060), the MedTech Centre of Research Expertise, University of Canterbury, New Zealand and Monash University Malaysia Advance Engineering Platform (AEP) for supporting this research.

Conflict of Interest Declaration

This study received a research grant (IF0219I1060) from the Ministry of Energy, Science, Technology, Environment and Climate Change (MESTECC).

References

[1] E. Emrath, The Basics of Ventilator Waveforms, Current Pediatrics Reports 9 (1) (2021/03/01 2021,) 11–19, https://doi.org/10.1007/s40124-020-00235-4.
[2] E. Kipnis *et al.*, "Monitoring in the Intensive Care," *Critical Care Research and Practice,* vol. 2012, p. 473507, 2012/08/27 2012, doi: 10.1155/2012/473507.
[3] P. B. Blanch and M. J. Banner, "A new respiratory monitor that enables accurate measurement of work of breathing: A validation study," *Respiratory Care,* Article vol. 39, no. 9, pp. 897-905, 1994. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0027984746&partnerID=40&md5=74b6b4dbd3646658fd5b7dedbf267a08.
[4] L. Blanch, B. Sales, J. Montanya, U. Lucangelo, O. Garcia-Esquirol, A. Villagra, E. Chacon, A. Estruga, M. Borelli, M.J. Burgueño, J.C. Oliva, R. Fernandez, J. Villar, R. Kacmarek, G. Murias, Validation of the Better Care® system to detect ineffective efforts during expiration in mechanically ventilated patients: a pilot study, Intensive Care Medicine 38 (5) (2012) 772–780.
[5] R. Kohli-Seth, J.M. Oropello, THE FUTURE OF BEDSIDE MONITORING, Critical Care Clinics 16 (4) (2000) 557–578, https://doi.org/10.1016/S0749-0704(05)70134-2.
[6] M. S. Jassas, A. A. Qasem, and Q. H. Mahmoud, "A smart system connecting e-health sensors and the cloud," in *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 3-6 May 2015 2015, pp. 712-716, 10.1109/CCECE.2015.7129362.
[7] M. S. D. Gupta, V. Patchava, and V. Menezes, "Healthcare based on IoT using Raspberry Pi," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 8-10 Oct. 2015 2015, pp. 796-799, 10.1109/ICGCIoT.2015.7380571.
[8] R. Dudas, C. VandenBussche, A. Baras, S.Z. Ali, M.T. Olson, Inexpensive telecytology solutions that use the Raspberry Pi and the iPhone, Journal of the American Society of Cytopathology 3 (1) (2014) 49–55.
[9] A. M. Milenković, I. M. Marković, D. S. Janković, and P. J. Rajković, "Using of Raspberry Pi for data acquisition from biochemical analyzers," in *2013 11th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, 16-19 Oct. 2013 2013, vol. 02, pp. 389-392, doi: 10.1109/TELSKS.2013.6704405.
[10] S. G. Mansfield, "PIMAP: A System Framework For Patient Monitoring," ed, 2021.
[11] G. B. Rehm *et al.*, "Development of a research-oriented system for collecting mechanical ventilator waveform data," *Journal of the American Medical Informatics Association,* vol. 25, no. 3, pp. 295-299, 2018, 10.1093/jamia/ocx116.

[12] E.J. van Drunen, Y.S. Chiew, C. Pretty, G.M. Shaw, B. Lambermont, N. Janssen, J.G. Chase, T. Desaive, Visualisation of time-varying respiratory system elastance in experimental ARDS animal models, BMC Pulmonary Medicine 14 (1) (2014), https://doi.org/10.1186/1471-2466-14-33.
[13] U. Lucangelo, F. Bernabè, L. Blanch, Lung mechanics at the bedside: make it simple, Curr Opin Crit Care 13 (1) (2007) 64–72, https://doi.org/10.1097/MCC.0b013e32801162df.
[14] S.E. Morton et al, Optimising mechanical ventilation through model-based methods and automation, Annual Reviews in Control (2019), https://doi.org/10.1016/j.arcontrol.2019.05.001.
[15] Q. Sun *et al.*, "Over-distension prediction via hysteresis loop analysis and patient-specific basis functions in a virtual patient model," *Computers in Biology and Medicine,* p. 105022, 2021.
[16] C. Zhou, J.G. Chase, J. Knopp, Q. Sun, M. Tawhai, K. Möller, S.J. Heines, D.C. Bergmans, G.M. Shaw, T. Desaive, Virtual patients for mechanical ventilation in the intensive care unit, Computer Methods and Programs in Biomedicine 199 (2021) 105912.
[17] N.L. Loo, Y.S. Chiew, C.P. Tan, M.B. Mat-Nor, A.M. Ralib, A machine learning approach to assess magnitude of asynchrony breathing, Biomedical Signal Processing and Control 66 (2021), https://doi.org/10.1016/j.bspc.2021.102505 102505.
[18] C.Y.S. Ang, Y.S. Chiew, L.H. Vu, M.E. Cove, Quantification of respiratory effort magnitude in spontaneous breathing patients using Convolutional Autoencoders, Computer Methods and Programs in Biomedicine 215 (2022), https://doi.org/10.1016/j.cmpb.2021.106601 106601.
[19] T.C. Chong, N.L. Loo, Y.S. Chiew, M.B. Mat-Nor, A.M. Ralib, Classification Patient-Ventilator Asynchrony with Dual-Input Convolutional Neural Network, IFAC-PapersOnLine 54 (15) (2021) 322–327, https://doi.org/10.1016/j.ifacol.2021.10.276.
[20] J. W. W. Lee *et al.*, "Stochastic Modelling of Respiratory System Elastance for Mechanically Ventilated Respiratory Failure Patients," (in eng), *Ann Biomed Eng,* pp. 1-16, Aug 25 2021, 10.1007/s10439-021-02854-4.
[21] Y. Marchuk *et al.*, "Predicting Patient-ventilator Asynchronies with Hidden Markov Models," (in eng), *Sci Rep,* vol. 8, no. 1, p. 17614, Dec 4 2018, 10.1038/s41598-018-36011-0.

QING ARN NG received his B.S. degree in mechanical engineering and M.Sc. degree in engineering science (research) from Monash University in 2020 and 2022, respectively. His interests include modelling and analysis of systems, with focus on biomedical engineering.