

PROJECT REPORT

In Medical Engineering & eHealth

Physicalisation of Upper Airway Models for Applications in Respiratory Research

By: Leonhard Hauptfeld

Student Number: me21m003

Supervisor: Richard Pasteka

Wien, September 28, 2022

Contents

1	Introduction	1
1.1	Vision	1
1.2	Goal of the project	1
1.3	State of the art	1
1.4	Analysis	2
2	Materials and Methods	2
2.1	Requirements and approaches for a physicalisation process	2
2.1.1	Requirements	2
2.1.2	Evaluating software	3
2.1.3	Process development	4
2.2	Requirements and approaches for a printing process	5
2.2.1	Requirements	5
2.2.2	Process development	5
2.3	Documentation of processes	6
2.4	Developing and evaluating hardware and software for measurements	6
2.4.1	Hardware	6
2.4.2	Software	6
3	Results	7
3.1	Novel mesh editing processes	7
3.1.1	Software Evaluation	7
3.1.2	Hollowing	8
3.1.3	Cutting into segments	8
3.2	Parametric connector system	9
3.2.1	Adapter plate	10
3.2.2	Features	12
3.3	Printing process	12
3.4	UAM Guide - Guide document for airway work	13
3.5	UAM Thing - A thing for airway measurements	14
3.5.1	Choosing components	14
3.5.2	Developing circuit board and case	15
3.5.3	Implementing firmware	17
3.5.4	Preliminary testing	17
3.6	UAM Workbench - Software for airway work	19

3.7 Evaluation of entire ecosystem	19
4 Discussion	20
4.1 Processes and ecosystem evaluation	21
4.2 Shortcomings and possible improvements of UAM Thing	21
4.3 Further process development	22
5 References	23
List of Figures	24
List of Tables	24
List of Code	24

1 Introduction

1.1 Vision

Upper airway modelling is a relatively new field and is being widely researched nowadays. Among other research, it could even gain relevance with regards to the recent proliferation of the SARS-CoV-2 variants, as the virus increasingly attacks the upper airways[1]. Understanding and modelling these passages provides insight into their functionality and the effect of diseases. Aside from maintaining the confidentiality of volunteer CT data, there are also no significant ethical concerns with this sort of modelling, unlike in the case of animal testing. Some work has already been done in this project by a student group - the previous "Upper Airway Model" project succeeded in providing a 3D airway model from a single CT scan that can be rapidly manufactured for testing. However, the variability of human anatomy through age and other factors means that only one model is not nearly enough to account for the wide range of possible airway configurations and as such the process had to be altered and refined to allow for rapid creation of new manufacturable airway models to consequently ensure more meaningful results through a broader range of tested models.

1.2 Goal of the project

The goal of the project was to turn a digital model of an upper airway system to a physical one for testing. The physical model was to be tested using pressurized air and measurements taken that prove soundness and air-tightness of the model. For the physical testing, hardware and software that could take those measurements needed to be developed. To facilitate process usability, a document explaining the streamlined process was to be created.

1.3 State of the art

Some upper airway models exist at UAS Technikum Wien for researching mask effectiveness, inhaler efficiency and other respiration related topics. No guidelines or processes for development of models exists or was accessible for the purposes of this research, therefore none is assumed to exist.

1.4 Analysis

The lack of a more universally applicable and documented process needs to be addressed. In addition, development of an in-house measurement device with accompanying custom software could benefit research by removing the need for expensive external equipment. Furthermore, such a device could be customized to the needs of research conducted at UAS Technikum.

2 Materials and Methods

The main processes are split into those required to accomplish creating a model for physical testing ("Physicalisation") and those necessary for printing it. The resulting processes then need to be documented. Finally, a device and accompanying software for physical measurements are developed.

2.1 Requirements and approaches for a physicalisation process

2.1.1 Requirements

The process of editing the mesh for physicalisation needs to keep to a few key requirements.

- **Accuracy** | The resulting mesh needs to be true to the original airways with regards to analysis.
 - The air-facing parts of the resulting mesh for printing needs to match the air-facing parts of the original mesh.
 - Parts that don't interact with airflow (for example, the outer wall) can be modified as required.
- **Usability** | The process needs to be as easy to use as possible by someone with little previous knowledge.
 - Keep number of steps to the minimum necessary
 - Document common errors and anomalies with ways to resolve them
- **Universality** | The process needs to be as universally applicable as can be.
 - The basic principles should be applicable to the majority of airways.

- There cannot be a catch-all solution to every airway model. Universality is restricted in that regard.

2.1.2 Evaluating software

The process will need to make use of one more applications to aid in mesh manipulation. The amount of required software should be kept to a minimum to avoid the user needing to install it all in addition to keeping it updated.

Software is evaluated based on a few key criteria:

- **Ease of Use** | An overall impression of the UI, how "fool-proof" the software is, how many errors it allows, how fast it can be learned, etc.
- **Functionality** | How many tools the software provides, how useful they are for this application and how well they work
- **Availability** | Whether the software is free to download, free for academic purposes or requires a purchase. Additionally, being Open Source is a bonus due to the guaranteed continued availability.
- **Automatability** | A minor point - how automatable the software is through scripting or similar. Some processes could eventually be automated through this.

The evaluated software will be Autodesk Fusion 360, Blender and OpenSCAD.

2.1.3 Process development

The process is to be developed by first identifying all the required subprocesses and then working out a basic working routine for each. The basic routine is then evaluated based on the process evaluation criteria - if they are not deemed sufficiently fulfilled (e.g. a significant deviation from the original mesh) the existing steps are then refined by experimenting with new tools and methods. The new steps are then again evaluated. This repeats until all the criteria are sufficiently fulfilled and the process is worked out (Fig. 1).

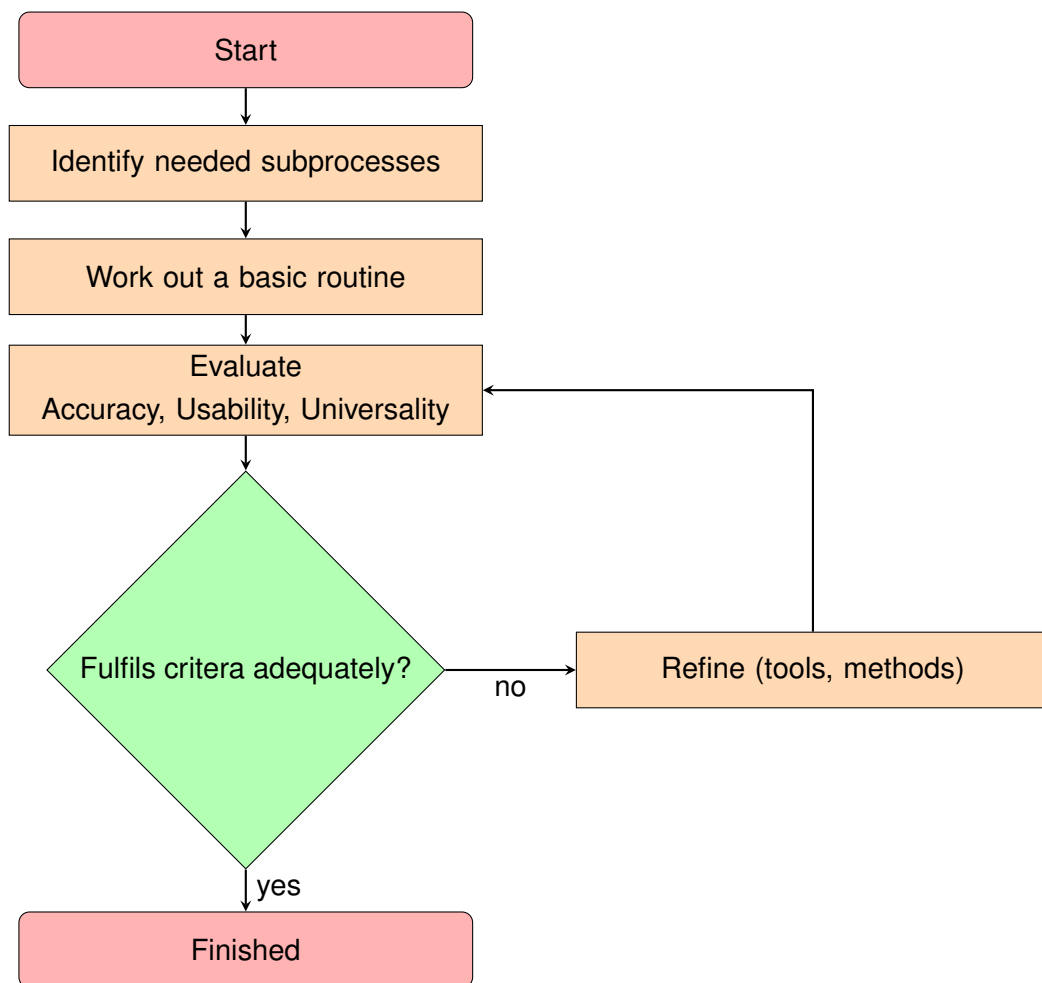


Figure 1: Physicalisation process development

2.2 Requirements and approaches for a printing process

2.2.1 Requirements

The requirements to the printing process are broadly the same as to the mesh process - Accuracy, Usability and Universality. Accuracy in 3D printing pertains to how accurate the printed airway model will be compared to the provided mesh. Universality is additionally relevant when it comes to the wide variety of available 3D printers and how well the printing process works for them. The usability concerns remain the same. As an additional requirement, the printed model will need to be air tight.

2.2.2 Process development

As most of the printing process is automated, the relevant user input is in material choice and printing settings. Different materials and printing settings have to be tried and the printed results evaluated - again with regards to accuracy and universality.

Materials to be evaluated:

- **PLA** | Brittle but easy to work with, can be printed on any FDM printer
- **ABS** | Stronger and more flexible but requires more advanced FDM printers and is prone to deforming during print
- **TPU** | Strong and flexible, provides good sealing surfaces, hardest to print of all three

Settings to be evaluated:

- **Layer Height** | Height of an individual FDM layer, 0.1, 0.15 or 0.2mm
- **Supports** | Where to generate support struts for model overhangs
- **Model Placement** | Where to place and how to orient the model on the print bed

All other settings are too specific to individual printer models to be set with any certainty. The prints will be tested on an Ultimaker 2+.

Models are printed using a subset of settings and evaluated based on the above-mentioned criteria. If a printed model does not pass the criteria (e.g. is not air tight or too brittle) the settings are changed and the model is re-printed to be evaluated again. If a printed model passes muster, the settings are noted down for documentation.

2.3 Documentation of processes

Both the finalized mesh modification and printing processes are to be documented in an easily accessible form with picture guidance. This resulting guide document needs to conform to the same usability requirement as the processes themselves.

2.4 Developing and evaluating hardware and software for measurements

2.4.1 Hardware

For measurements of airflow and pressure in physical airway models, a device needs to be created that can take these measurements. The requirements are as follows:

- **Flow Measurements** | Take airflow measurements from 2 or more sensors on a typical human breathing scale (5-10 l/min)
- **Pressure Measurements** | Take air pressure measurements from 2 or more sensors on a typical human breathing scale (0-10kPa)
- **Transmission** | Transmit those signals in a timely and reliable manner to a computer

An appropriate microcontroller and sensors that work within the specified ranges need to be selected. In addition, a method of data transmission needs to be worked out that adheres to the timeliness and reliability principles.

2.4.2 Software

The software needs to be able to handle the incoming data from the measurement device and log it to storage reliably. Additional functionality could include display and analysis but these features are not required.

A language and/or framework that can handle the requirements and is maintainable needs to be selected and the program implemented.

3 Results

3.1 Novel mesh editing processes

3.1.1 Software Evaluation

Table 1: Mesh software evaluation

Software	Ease of Use	Functionality	Availability	Automatability
Fusion 360	Centered around usability. Friendly user interface, easiest CAD software.	Strong CAD and Mesh functionality, centered around physical models.	Free for Education, Proprietary	None
Blender	Medium usability. Lots of tutorials are available, can be learned in a few days.	Primarily used for CGI. Some 3D printing tools are available. Wide array of mesh tools.	Free, Open Source	Python Interface
OpenSCAD	Not easy. Models are created in custom scripting language. Steep learning curve.	CAD modelling only. Almost limitless programmable functionality.	Free, Open Source	Completely (Models are Scripts)

Blender was chosen for mesh manipulation and OpenSCAD for CAD modelling of required additional parts. Blenders powerful mesh editing tools and OpenSCADs versatility in addition to both of these applications being Open Source were key decision points.

3.1.2 Hollowing

The Blender "Solidify" modifier was used to create an airway mesh with a wall. By specifying a negative value for wall thickness, the walls were created towards the outside of the airways and thus did not narrow the inner airway paths (Fig. 2) Inside geometry was ignored by the tool, so inside features like tonsils were unaffected.

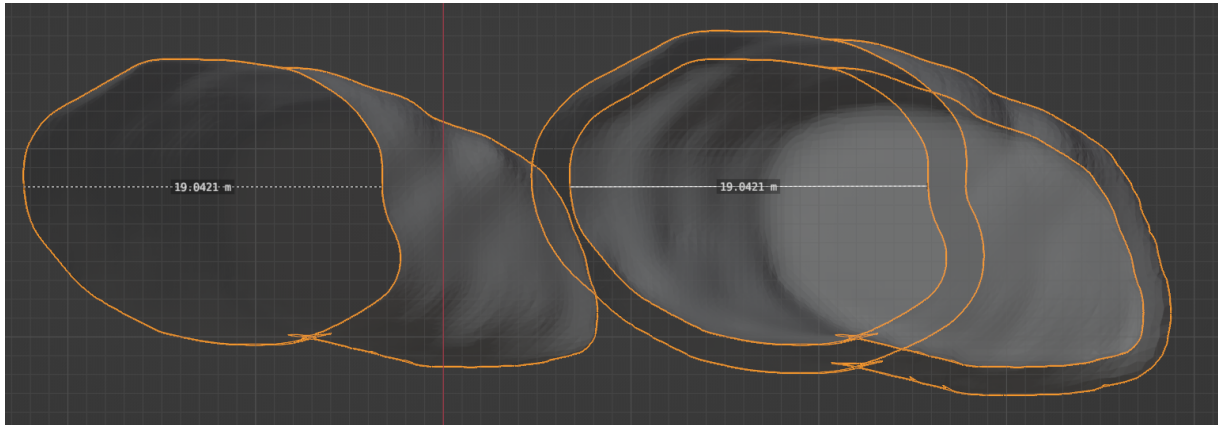


Figure 2: Hollowing in Blender

3.1.3 Cutting into segments

Two different Blender tools for mesh slicing - the knife and bisect tools - were tested. The bisect tool was used for the final process.

Knife Tool

The knife tool was the first method attempted for cutting airway meshes into segments. It is a fairly advanced tool that allows projecting 2D shapes onto the 3D geometry to create cut points. While powerful, this tool requires many steps to use - 2D geometry needs to be created and the projection needs to be adjusted accordingly. Since separation of airway segments requires just a cut along a plane, this tool failed on the Usability criterion.

Bisect Tool

The bisect tool is a fairly straightforward tool in Blender to cut a mesh along an infinite plane. It is extremely intuitive to use (dragging a line over the mesh). The tool itself only creates the vertices and edges along the mesh - in addition, the mesh needs to be separated. This is accomplished using the "Rip" tool. (Fig. 3)

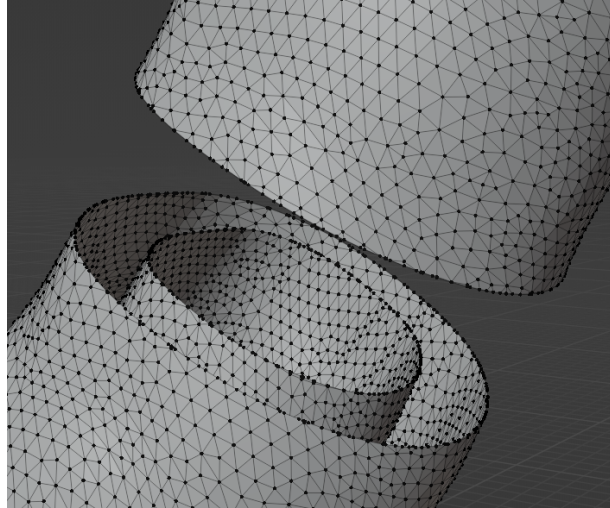


Figure 3: Bisecting in Blender

3.2 Parametric connector system

Once an airway model is hollowed and cut into segments, there needs to be a way to connect the different segments after printing in an air-tight way. Additionally, there need to be measurement ports for flow (input/output points of the model) and pressure.

These requirements are accomplished by a universal connector system that consists of adapter plates and features. Adapter plates are parametric round, flat surfaces with a large center hole on the model. These are attached at the end of airway segments and allow them to be connected to each other. Adapter plates can also be placed on the side of a model.

Apart from to each other, adapter plates can also be connected to features. Features are adapter plates that have certain attachments that fulfil a specific purpose like providing a spot to attach a ventilator tube or pressure sensor. The core concept of the system is that adapter plates and features generated with the same parameters are always compatible with each other. As long as a model has the parameters of its adapter plates documented, matching features can be generated for it.

The adapter plates and features are implemented in OpenSCAD scripts.

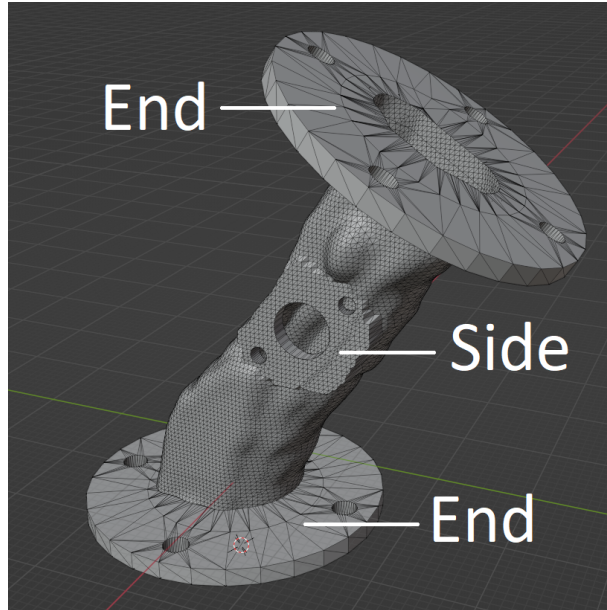


Figure 4: Illustration of the connector system

3.2.1 Adapter plate

Adapter plates are simple discs of a certain thickness with a large centered hole (Fig. 5). They can be generated either as the disk for attachment to a mesh or in a drill form that is used for inserting appropriately spaced holes into an existing flat surface on a mesh.

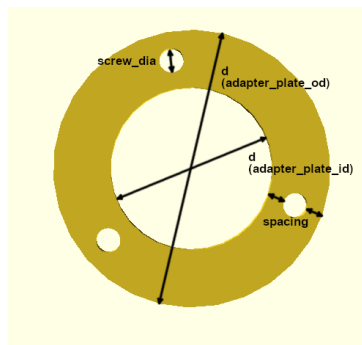


Figure 5: Adapter plate

They are customizable through the following parameters:

Table 2: Adapter plate parameters

Name	Variable	Default	Description
Diameter	d	20	Either the inner or the outer diameter of the adapter plate, depending on if adapter_plate_id or adapter_plate_od is used.
Height	h	2	The thickness of the adapter plate or the length of the drill.
Spacing	spacing	2	Space between screw holes and adapter edges.
Screw Diameter	screw_dia	3	Diameter of screw holes. It is generally advisable to keep this a bit below the actual screw size for a good fit, e.g. 2.8 for M3 bolts.
Screw Number	screw_num	3	Number of screw holes evenly distributed across the plate. It is not advisable to drop this number below 3 unless space is tight.

Since default values for all the parameters are provided, only differing values have to be documented for adapter plates on models.

Attaching an adapter plate to a segment end

After cutting a segment of the airway mesh, the geometry is left open (Fig. 3). An adapter plate needs to be attached and this open geometry closed (made uniform) to allow for error-free 3D printing.

The diameter of the open airway segment is measured. Then, an adapter plate with the appropriate inner diameter is generated. It is moved into position to align with the cut and joined with the surrounding mesh. The exact process with images can be found in the guide document.

This process went through several iterations of refinement, reducing the number of steps and most importantly automating the most time-intensive step (selecting the specific mesh for removal) by use of a built-in Blender tool. This also cut down dramatically on the number of user errors that could occur, since the manual selection of specific mesh also required the most manual precision.

Attaching an adapter plate to a segment side

For an adapter plate surface on the side of an airway segment, the Blender knife tool is used. Unlike in cutting of airway segments, this process requires the precision of that tool. A circular segment is cut out of the side of the mesh and brought to be vertically level. A "drill" adapter plate object is generated and boolean combined with the airway mesh to create matching screw holes for attachment of a feature. The exact process with images can be found in the guide document.

This process went through one major iteration where the process wasn't made easier for the user but the resulting mesh ended up far cleaner and easier to 3D print by removing jagged edges on the corners of the flat surface.

3.2.2 Features

There are two features that can be generated for adapter plate attachment points - air hose and vent tube connectors.

- **Air hose connectors** are a design that allows for attachment of any size air hose, for example to attach a small diameter hose for a pressure sensor.
- **Vent tube connectors** are specifically made to provide a tight fit for standard 21/22mm ventilator tubing.

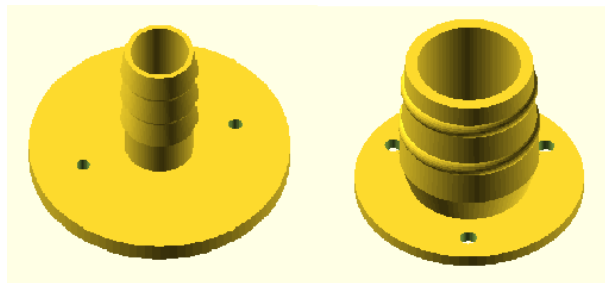


Figure 6: Air hose (left) and vent tube (right) connector features

3.3 Printing process

Printing settings were evaluated by modelling and attempting to print trachea and larynx/pharynx of the model with the identifier "0190673870_16a_M". Only one set of settings was tried and provided adequate results. Only one material was tried (PLA).

A layer height between 0.15mm and 0.2mm is acceptable for printing an airway model. The increased speed from these layer heights dramatically cuts down printing times and the added

precision provided by printing with lower layer heights provides no advantages considering the low resolution of the underlying CT data.

Support material has to be generated everywhere. With a quality 3D printer, the overhang angle at which such material is generated can be reduced. Due to the model being split up into many segments, it is not a problem to remove support material from inside the airways with a screwdriver or long, slim pair of pliers.

Models should be placed with one of their flat adapter plates facing the buildplate. This provides maximum adhesion to the buildplate. Whichever plate facing down requires less support material is to be preferred.



Figure 7: Printed larynx/pharynx of 0190673870_16a_M

3.4 UAM Guide - Guide document for airway work

The finalized processes for physicalising and printing were documented in an appealing format by use of the tool mkdocs. Step-by-step guides with pictures written in markdown are converted a full HTML page that can be considered a wiki of upper airway modelling. The guide can also be exported as a PDF document.

Error states were documented in additional "info boxes" that are inserted into the guide when-

ever appropriate to warn of potential common errors or additional information such as useful tools or suggested procedures.

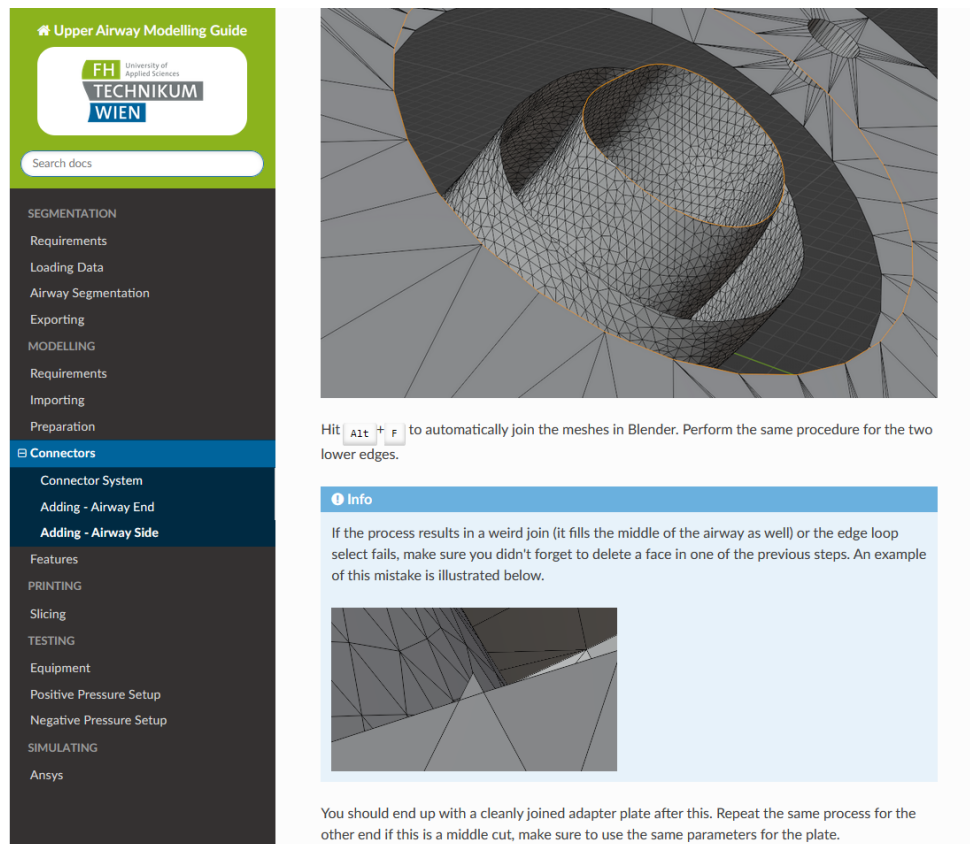


Figure 8: UAM Guide

3.5 UAM Thing - A thing for airway measurements

UAM Thing is a sensor platform that supports four MPX5100/MPX5010 pressure sensors (P1-P4) and four SFMxxxx flow sensors (Q1-Q4). It transmits data via MQTT to a computer that can log the data via a tool of choice.

3.5.1 Choosing components

The Espressif ESP32 was chosen as the microcontroller at the heart of the device. It is fast enough to handle more concurrent operations than this project requires, has a crash-resistant real time operating system and extensive connectivity via WiFi, two I2C controllers and 2 ADCs with 6 channels each.

For a pressure sensor that could provide readings in the required ranges, the MPX5xxx series was chosen. The MPX2xxx series was evaluated but the need for an external amplification

circuit would've greatly increased circuit complexity. The MPX5xxx provides internal amplification for an output level between 0.2V and 4.7V. This would only require a voltage divider for the ESP32s maximum input level of 3.3V.

As a flow sensors, the SFMxxxx series was chosen due to it being made for ventilation devices and therefore supporting all the common tubing connections and measuring ranges. Its 5V logic level would require level shifting to the ESP32s 3.3V and the inability to change I2C adress on the SFM3xxx series meant multiplexing of the I2C bus would be required.

3.5.2 Developing circuit board and case

SFMxxxx

Due to the SFM3xxx series having a fixed I2C address (0xH40), the choice was to either use both of the ESP32s hardware I2C drivers, software I2C or an I2C multiplexing chip. Since software I2C is less reliable than the hardware driven equivalent and level shifting had to be done anyway, the multiplexing method was chosen. A breakout board based on the TCA9458A multiplexer chip is connected to all four SFM sensor channels. The ESP is connected to this multiplexer via one of its hardware I2C drivers. The ESP would have to continually iterate over all connected channels to get new measurement values. Since the I2C bus on all of these devices can operate at a high 400kHz and SFM sensors have a response time of 1ms (1kHz), this wouldn't impede sampling rate in any way. In addition, the TCA9458A handles level shifting between 3.3V and 5V. All I2C lines are pulled up to the appropriate voltage through 10kOhm resistors.

The SFM sensors feature a unique pad connector for which a clamp has to be manufactured that attaches to the sensor. A design from the online repository Thingiverse (4924085 by "nick-sears") was used as it could be manufactured with readily available pogo pins. The other end of the connection was to be made through 4 pin JST-XH connectors. The heating lines on the sensor are left unconnected, as the evaluations within the scope of this project did not require working with very humid air.

MPX5xxx

The MPX sensors provide a typical output voltage range from 0.2V to 4.7V. This was halved by use of a voltage divider ($R_1=100\text{k}\Omega$, $R_2=100\text{k}\Omega$). The resulting output voltage would therefore range from 0.1V to 2.35V. Set to 11dB of attenuation, the ESP32s ADC could reliably read voltages from 150mV to 2450mV (Fig. 9). Below 150mV, the ADC reads zero - the pressure ranges of 0 to 0.33kPa (MPX5010) and 0 to 3.33kPa (MPX5100) are therefore unreadable.

The sensors are connected via 6-pin 2.54mm pin headers. Since the MPX5100 and MPX5010 both share the same pin arrangement, they can be used interchangeably. This configuration is fully supported.

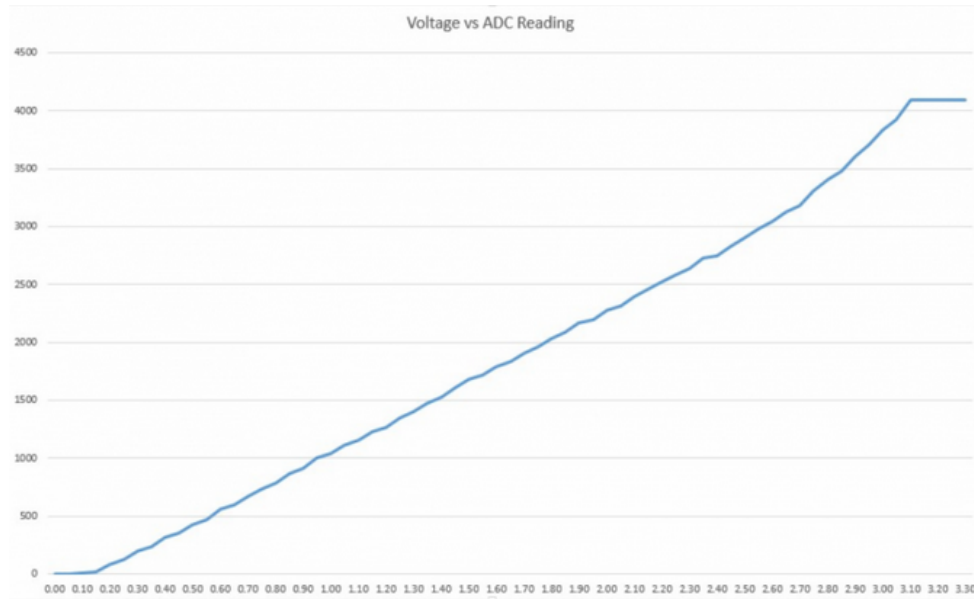


Figure 9: ESP32 ADC curve at 11dB attenuation

Smoothing capacitors are used for the power supply and each ADC channel as recommended by the application circuit in the MPX datasheets to provide more consistent results.

Hardware

The schematic was drawn up in KiCads eeschema and a hand-solderable board was created in KiCads pcbnew. A 3D model of this circuit board was then imported into OpenSCAD, where a case was modelled around it. The case features descriptions of the sensor ports and a little guide graphic to prevent insertion of the MPX sensors in the wrong orientation.

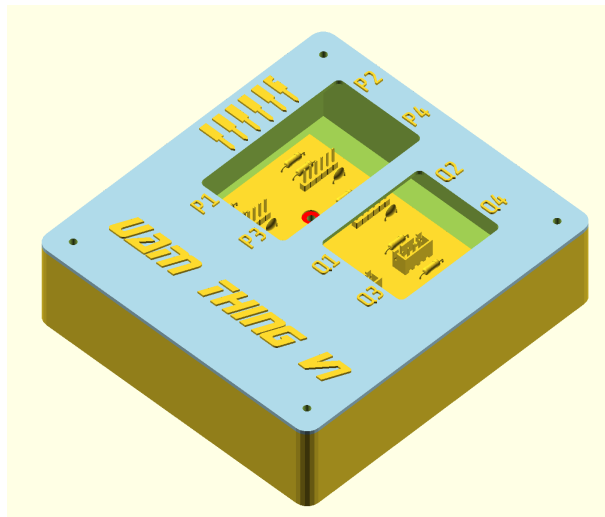


Figure 10: UAM Thing case

3.5.3 Implementing firmware

MQTT was chosen as the protocol for transmission of data. MQTT is based on publishing data in channels identified by strings and is simple, fast and widely implemented for all kinds of programming languages.

On Startup, the ADC channels are configured and the internal calibration values loaded to ensure correct readings from the MPX sensors. The SFM sensors are contacted through I2C and their serial as well as calibration values read. SFM sensors that don't respond on initialization are excluded from any further querying to save on unnecessary communication attempts.

The ESP32 then creates a WiFi network with a name configured in the firmware. As soon as a computer connects, it attempts to connect to an MQTT server on that computer. Publishing of sensor data on that server then starts immediately.

Two tasks in the ESP32's RTOS then run periodically every 200ms.

- The FlowSensors task iterates through all connected SFM sensors and reads their values. The values are converted to slm and reported on the MQTT channel `"/uam-thing/<device-id>/flow/<sensor-id>"`. Additionally, any communication with SFM sensors is error checked by means of CRC8 values to ensure accurate reporting of sensor data.
- The PressureSensors task iterates over all ADC1 channels and reads the raw value of each. Using the internal calibration of the ESP32, the readings are converted to voltages where they are converted to kPa. The ratio is determined by whether the port is set to be an MPX5010 or MPX5100 sensor in the firmware. The kPa values are then reported on the MQTT channel `"/uam-thing/<device-id>/pressure/<sensor-id>"`.

3.5.4 Preliminary testing

A prototype UAM Thing was constructed.

To establish plausibility of pressure sensor readings, an MPX5100 sensor was attached to a pressure source set to 0.5 bar or 50kPa. The sensor readings read between 49 and 52kPa. As the pressure source manometer was not highly accurate, this was determined to be acceptable to establish basic plausibility of sensor readings. A separate evaluation against a professional pressure measurement device would need to be performed to perform a full calibration.

Since the SFM sensors are pre-calibrated and communicate via I2C, a basic plausibility test was not considered necessary.



Figure 11: Pressure testing setup

3.6 UAM Workbench - Software for airway work

Creation of airway models is a multimodal environment, where CT segmentation, CAD, modelling and datalogging tools have to be used. To create a data structure that could house all that data and bind it to the context of one model, a software called UAM Workbench was prototyped. This software would store all metadata pertaining to an airway model in a JSON file. Remote interfaces to 3D Slicer and Blender were prototyped to allow the software to aid in tasks within these programs and help it manage their complex file structure. In addition, it was to contain a built-in MQTT server for UAM Thing and a tool for logging and displaying its sensor data.

This app would be developed using the Tauri development framework to ensure a modern user experience.

Basic management of the file structure as well as prototype interfaces to Blender and UAM Thing were achieved. The tool exists in a little showcase format but is, as of submission (28.9.2022), not production-ready.

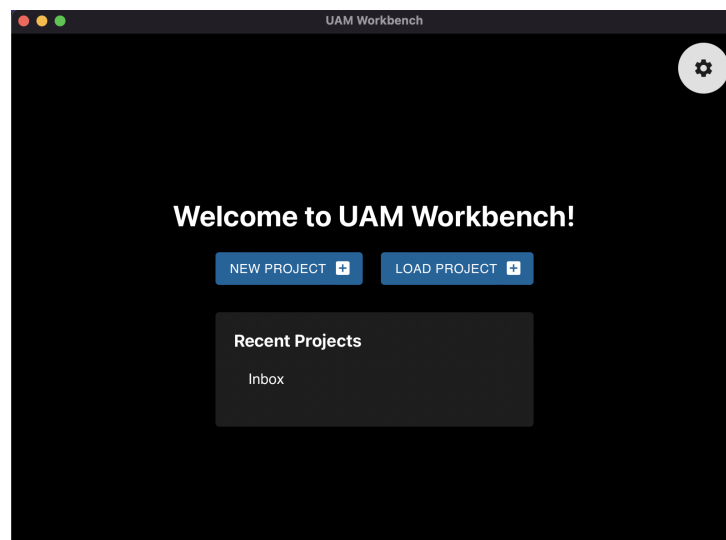


Figure 12: UAM Workbench - Welcome Screen

3.7 Evaluation of entire ecosystem

To evaluate the entire ecosystem (creation process, connector system, printing process & UAM Thing) a model of the dataset "0190673870_16a_M" was constructed and a trachea and larynx/pharynx printed. The trachea had vent tube connector features attached at both ends and an air hose connector feature for 6mm inner diameter tubing attached on the side. No gaskets or other sealing methods were used between adapter plates and features.

An SFM3300 flow sensor at the inlet was connected to port Q1 of the UAM Thing, another SFM3300 sensor was attached at the outlet of the trachea and to port Q2 of the UAM Thing. An MPX5010DP pressure sensor was connected to port P1 of UAM Thing and a tube was

connected between the side feature of the trachea and the positive pressure input. Figures 13 and 14 illustrate this setup.

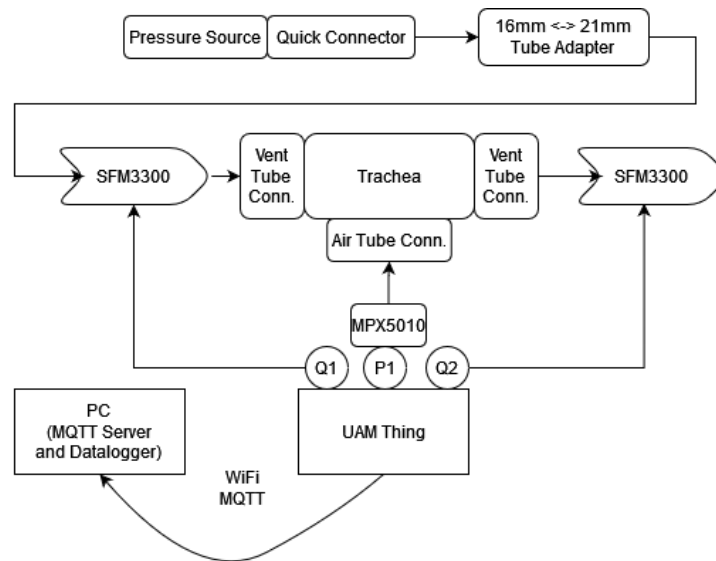


Figure 13: Ecosystem evaluation - Schematic

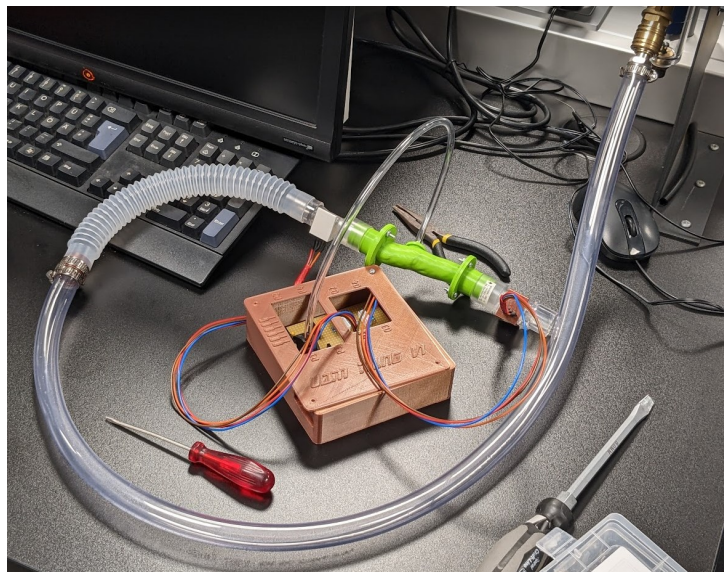


Figure 14: Ecosystem evaluation - Setup

Data was logged for a 25 second period while flow was periodically increased up to a point of 300 slm as measured by the inlet sensor Q1. A significant difference in flow between Q1 and Q2 would indicate a leak.

The variance in slm between Q1 and Q2 is negligible, on the order of 1slm. No significant leaks were observed. The pressure sensor remained mostly idle, only showing 1kPa on occasion when flows over 120 slm were present.

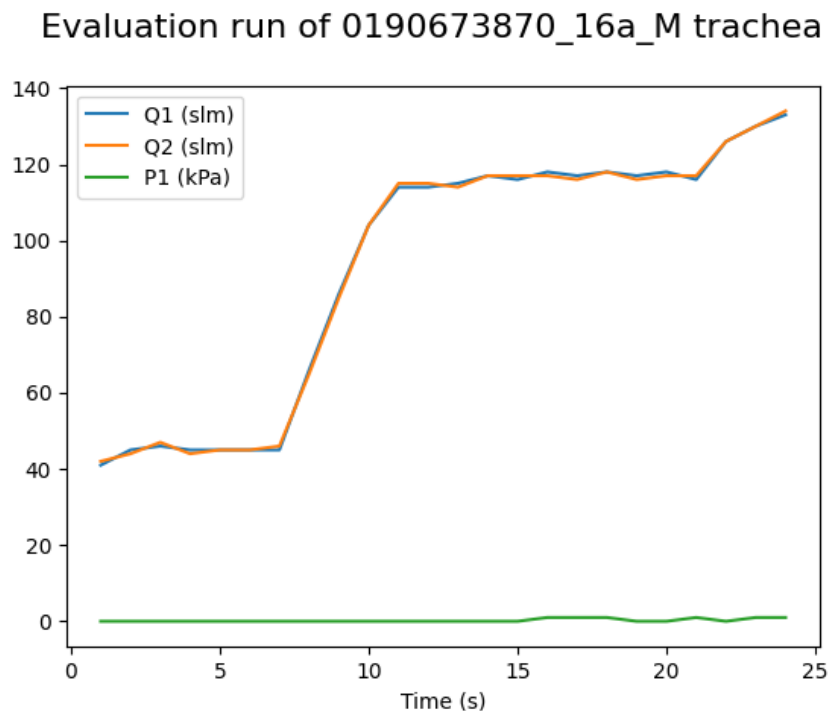


Figure 15: Ecosystem evaluation - Result

4 Discussion

4.1 Processes and ecosystem evaluation

Overall, the new processes and guide documents proved to be a success, as they were used quite extensively while conducting this very research. Error rates and modelling times were gradually reduced overall, though a proper case study about this including people without prior experience would have to be conducted to say this for certain.

The result in the ecosystem evaluation came unexpected, as a leak without any sealing measures was expected. The adapters performed better than expected, which might be attributable to their large surface area and the clamping force of the screws that hold them together. Whether they hold up in higher pressure scenarios would need further evaluation.

4.2 Shortcomings and possible improvements of UAM Thing

- A printed circuit board would provide a more robust electrical foundation and cut down on soldering work

- The pin headers proved to be insufficient in supporting the weight of the MPX sensors and attached tubing, eventual failure is probable. They could be supported through their own screw holes with a more advanced case design.
- Similarly, the JST connectors could be swapped for a sturdier alternative like DIN plugs. JST connectors are rated for around 10.000 cycles.
- More calibration work has to be done to ensure accuracy of the MPX sensors especially
- Since the device is powered off USB, a more stable 5V supply voltage would be advantageous to prevent crashes of the SFM sensors and ensure accuracy of the MPX sensors. A boost converter could accomplish this.

4.3 Further process development

- The physicalisation process needs to be trialed with more complex and non-uniform meshes. Some strange errors can occur in those cases, for which resolutions should be documented.
- More printing materials and technologies (SLA) need to be trialed with airway models.

5 References

- [1] M. Kozlov. “Omicron’s feeble attack on the lungs could make it less dangerous”. In: *Nature* 601 (Jan. 2022), p. 177. URL: <https://www.nature.com/articles/d41586-022-00007-8>.

List of Figures

Figure 1	Physicalisation process development	4
Figure 2	Hollowing in Blender	8
Figure 3	Bisecting in Blender	9
Figure 4	Illustration of the connector system	10
Figure 5	Adapter plate	10
Figure 6	Air hose (left) and vent tube (right) connector features	12
Figure 7	Printed larynx/pharynx of 0190673870_16a_M	13
Figure 8	UAM Guide	14
Figure 9	ESP32 ADC curve at 11dB attenuation	16
Figure 10	UAM Thing case	16
Figure 11	Pressure testing setup	18
Figure 12	UAM Workbench - Welcome Screen	19
Figure 13	Ecosystem evaluation - Schematic	20
Figure 14	Ecosystem evaluation - Setup	20
Figure 15	Ecosystem evaluation - Result	21

List of Tables

Table 1	Mesh software evaluation	7
Table 2	Adapter plate parameters	11

List of Code