

NAME PEGGY PEI
MATRICULATION NUMBER S2134810
MENG HONS PROJECT PHASE 2 REPORT
A PORTABLE BIO-SENSING DEVICE- MEASURING
THE RESPIRATORY IMPEDANCE FOR COPD
PATIENTS
17 JANUARY 2023

MEng Project Mission Statement

A portable bio-sensing device- Measuring the respiratory impedance for Chronic obstructive pulmonary disease (COPD) patients

Student: Peggy Pei (s2134810)

Supervisor: Dr Stewart Smith

Subject Area: Biomedical application/ Biosensor/ Micro-control system

Project Definition:

Chronic obstructive pulmonary disease(COPD) is the name for a group of lung conditions that cause breathing difficulties [1]. Patients suffering from this disease need regular monitoring on their breathing condition. The device based on spirometry remains significant for wide clinical use in this area. However, massive inconvenience has risen due to its inflexibility of gaint size and high requirement of cooperation from patients. Therefore, a portable biosensor device using the **Forced Oscillation Technique (FOT)** would be developed in this project as an approachable daily measurement for COPD patients. Generally, it will contain a sensor system to capture information provided by patients, a micro-control system to manage data feedback, and a communication system that can send information to the PC and preserve it for doctors and patients to check.

Task Clarification:

No.	Stage	Task
1	Micro-control system design (Phase 1)	Get familiar with STM32 MCU board and software. Design a micro-control system and communication system Run simulation
2	Sensing system design (Phase 1)	Figure out dynamics of airflow Design and fabricate a pneumotachometer Collect signals from sensing system(P/F)
3	Fabrication (Phase 2)	Combine systems together A model to imitate lung functions may be fabricated at this stage
4	Optimisation (Phase 2)	Additional functions as LED display will be extended here to improve the integrity of this biosensor system. Operate further signal processing, improve accuracy of the biosensor system.



Scope for Extension:

1. Thermal effect:

Previous research shows the accuracy of a pneumotachometer is mostly related to the temperature.[2] Therefore, a thermal sensor will be added to the control system in order to perform BTPS (Body temperature and pressure, saturated) corrections.

2. Signal processing:

Further biomedical signal processing should be added at phase-two stage to improve the accuracy for whole biosensor system.

3. Lung mimic:

An Extra model to intimate lung function can be developed and fabricated to prove the integrity of entire system.

Background Knowledge:

- Control and instrumentation
- Analogue/ digital signal processing
- Sensor and instrumentation
- Flow dynamics
- C-programming

Resources:

- STM32 Nucleo Board
- STM32 CUBEIDE
- Flow Head ML311
- Breadboard and jumper cables
- Digital potentiometer
- Temperature sensor
- Interface circuits
- Heating cables
- 3D printer
- Electromagnetic signal generator

Reference:

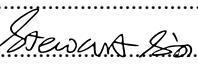
[1] [Chronic obstructive pulmonary disease \(COPD\) - NHS \(www.nhs.uk\)](https://www.nhs.uk) , cited on 14th Feb, 2022

[2] <https://www.pftforum.com/blog/pneumotach-accuracy/> ,cited on 8th Feb, 2022

The supervisor and student are satisfied that this project is suitable for performance and assessment in accordance with the guidelines of the course documentation.

Signed

Student:

Supervisor: 

Date: ..14/02/2022....

Abstract

Based on theories and simulations delivered at phase one, the work in phase two focused on the DVT(*design and verification test*) of the portable device, aiming to produce a reliable prototype for COPD(*Chronic obstructive pulmonary disease*) patients to conduct the respiratory measurement for daily monitoring. In addition to the detailed illustration of the DVT based on subsystem unit, the integrated prototype, including both mechanical and electronic output, was also delicately established in this report. One measurement was processed with the current prototype, which provides an unexpected result indicating the limitation of the present device. With analysis of this result, design constraints and improvements were carefully considered and outlined as potential future work. Although the intended measurement was not achieved at the end of phase two, cogitative discussions on both technical and business sides have been proposed here, expected to be a useful reference for the following students to make this device a further stage.

Declaration of Originality

I declare that this thesis is my
original work except where stated.

.....

Statement of Achievement

As an electronics and electrical engineering student, I had precious hands-on system design experience thanks to this MEng project. In phase two, my electronics skills related to PCB design, PCB manufacture, and real circuit test got well developed after my time at the lab for practical work during the last entire semester. Besides, I also designed a robust central control unit for this complex bio-sensing system through C programming, starting from zero. As a system design work, I obtained an opportunity to self-study the CAD design and 3D printed out the mechanism for the prototype myself eventually.

As an EEE student with great interest in the biotechnology field, I had a better understanding of the R&D process for medical devices. This project also drove me to think further about the meaning behind the technology in terms of the healthcare side. It did broaden my horizons and kindled my passion for continuously being part of the strength to develop technologies that can make people a quality life.

Contents

Mission Statement	i
Abstract	iii
Declaration of Originality	iv
Statement of Achievement	v
List of Symbols	viii
Glossary	ix
1 Introduction and Background	1
1.1 Project overview	1
1.2 Background	2
2 Design and Verification	5
2.1 Power System	5
2.1.1 Linear Voltage Regulator	5
2.1.2 Load Switch	7
2.2 Sensing System	8
2.2.1 Filter Designs and Simulations	9
2.3 Actuating System	12
2.3.1 Principle	12
2.3.2 Configuration	13
2.3.3 Verification	14
2.4 Heating System	16
2.4.1 Principle	16
2.4.2 Configuration	16
2.4.3 Verification	18
2.5 Communication System	19
2.5.1 GUI- TouchGFX	19

2.5.2	BLE model	23
2.6	Central Control System	24
2.6.1	Principle	24
2.6.2	Verification	32
3	Output and Result	34
3.1	Output	34
3.1.1	Electronic Output	34
3.1.2	Mechanical Output	39
3.1.3	Integration	39
3.2	Result	42
4	Future Work	44
4.1	Power System	44
4.2	Sensing System	45
4.3	Actuating System	45
4.4	Heating System	45
4.5	Communication System	45
4.6	Central Control System	46
5	Impact and Exploitation	47
5.1	Research and Development Process	47
5.2	Impact	49
Acknowledgements		51
References		53
A	Mechanical Drawings and Dimensions	54
B	PCB Assembly and Fabrication Files	57
C	Pinout view and System View	62
D	Cryptic configuration of the device	65

List of Symbols

ω	The angular velocity used for signals in the frequency domain.
j	Imaginary number.

Glossary

BLE Bluetooth Low Energy- a wireless communication protocol.

COPD Chronic obstructive pulmonary disease, a normal lung disease.

DVT Design and Verification Test.

FOT Forced Oscillation Technique.

LSA Linear solenoid actuator.

MCU Micro Controller Units.

USB-OTG-FS USB on the go, full speed.

Chapter 1

Introduction and Background

1.1 Project overview

This dissertation project of the MEng student Peggy Pei aims to design a portable biosensor device to measure the respiratory impedance for the **COPD** patients.

This report, based on the most of theories and preliminary design delivered at phase one, includes further **DVT** (*design and verification*) work for the biosensor system at phase two. As the device was sketched and designed from zero and only three months were allowed for project student to process all practical work, most of the initial DVT have been completed, however, follow-up tests related to device optimisation and result analysis are still hanging in the air and highly required for finally putting the device into the market. Therefore, instead of giving the number of pages on previous theories or subsequent analysis of results and optimisation approaches, the phase two thesis mainly focuses on demonstrating the process of DVT of this biosensor device.

The report starts with an overview introduction, providing academic background related to this project in both the physiology and technique field. Then detailed DVT process of each subsystem is explicitly illustrated afterward in chapter 2. Outputs and results measured by the current device are summarised in Chapter 3, which contains all achievements the device can approach at this stage. It is worth noting that a chapter named **Future Work** is added. This is because the project student fully understands design constraints and the complexity of the current system, any possible improvement has been carefully considered and shown in this chapter 4, rendered in subsystem units. The information provided might still not be sufficient and perfect, but hopefully, it could be a useful reference for follow-up students working on the same topic and trying to make this device to a further stage. Last but not least, the potential impacts carried by the current biosensor device in the respiratory measurement field are also carefully considered and explained in the final chapter 5, discussing the future exploitation and strategy that shall be contemplated to commercialise a medical device.

1.2 Background

COPD *Chronic obstructive pulmonary disease* is a chronic inflammatory lung disease, which is widely seen among people with a smoking history, polluted living environment, or inhaling other irritants that cause lung infections. Because of its chronic and pervasive nature, COPD patients need regular monitoring to make sure the condition is under control. However, current monitoring methods with professional equipment are complex and inconvenient as COPD patients have to come to the hospital and be examined by doctors with a large-size instrument. This is the drive of this project that a portable bio-sensing device is expected to be developed for COPD patients doing the daily measurement by themselves. More details about COPD can be found in the phase one thesis [1].

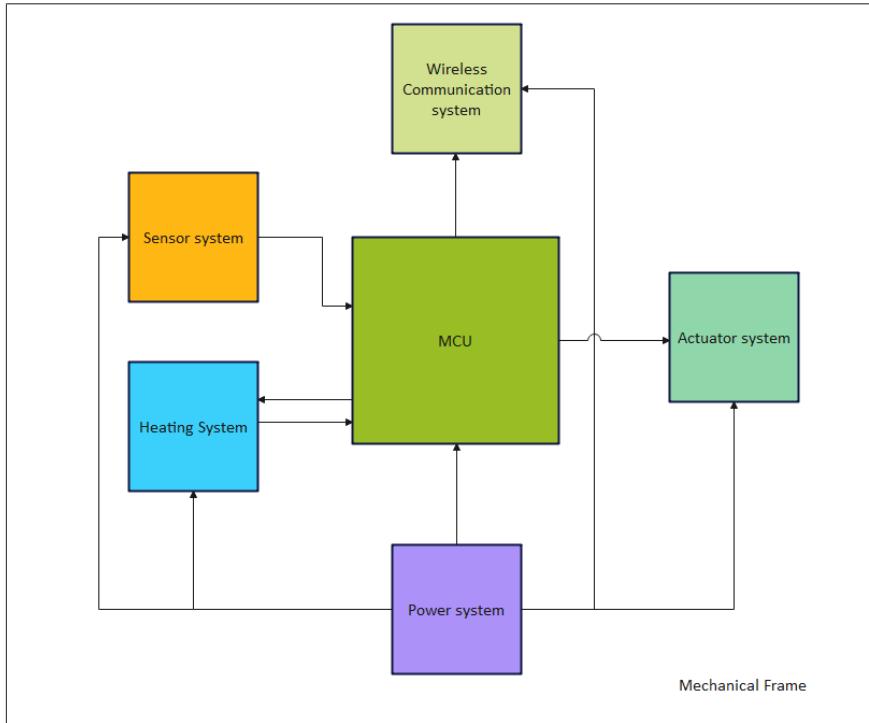
FOT *Forced oscillatory technique* is an advanced technique used to evaluate lung function. It measures the respiratory impedance of patients with a small actuating signal applied. This small actuating signal travels from the patient's mouth to their lung tissues, giving information for both large airways and peripheral airways. By measuring the pressure and flow rate of the patient's forced breath, the respiratory impedance can be calculated according to the equation 1.1.

$$Z(\omega) = R_{rs} + jX_{rs} = \frac{P(\omega)}{Q(\omega)} \quad (1.1)$$

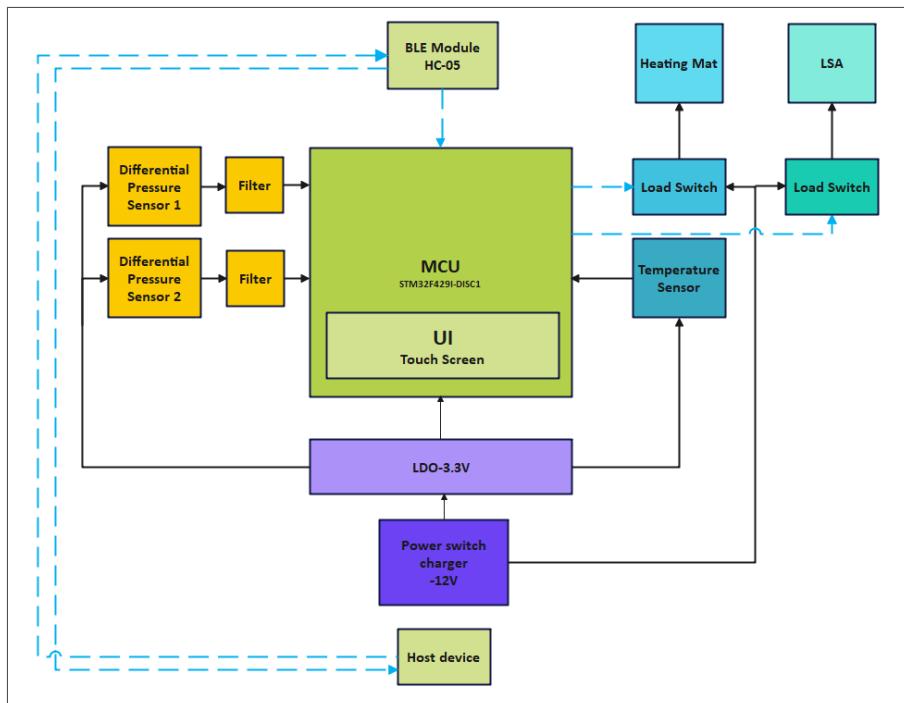
Detailed research such as advantages of **FOT**, parameters of FOT and differences between spirometry, FOT, IOS (*Impulse oscillatory system*) have been specifically illustrated in the phase one thesis [1], which will not be repeated here. In fact, the FOT provides high-accurate results for the respiratory measurement that is widely used for clinic application. This biosensor device was designed and developed fully resting on the FOT at both phase one and phase two stages.

System Diagram The system design was accomplished in phase one, delivering six function blocks of the bio-sensing system. The work at phase two still adhered to the original system block diagram as shown in figure 1.1a. However, detailed functional blocks of each subsystem were slightly modified compared to the preliminary design, which is shown in figure 1.1b. There are two modifications to the phase one design. Firstly, instead of using the push button IC to control the On/Off of the device, a more advanced touchscreen GUI (*Graphic User Interface*) was designed and implemented for users to engage with the device. What's more, instead of using the USB charging IC, a simpler 12V charger was applied for stable power input. More details of these modifications will be given when the DVT process of the communication system and the power system are illustrated, respectively.

Interpretation The operation process approximately maintains as same as the preliminary design at phase one. Detailed information such as the operating steps, parameters in measurement, and predicted value range can all be found on *Phase one thesis* [1]-> *Chapter 3: Methodology -> 3.4 Interpretation*, which would not be repeated again here. There is only one change that needs to be noted. Initially, the frequency of actuating signal was designed to sweep based on cycle counting, for example, frequency



(a) System block diagram of the biosensor device



(b) Functional block diagram of the biosensor device

Figure 1.1: The block diagram of biosensor system

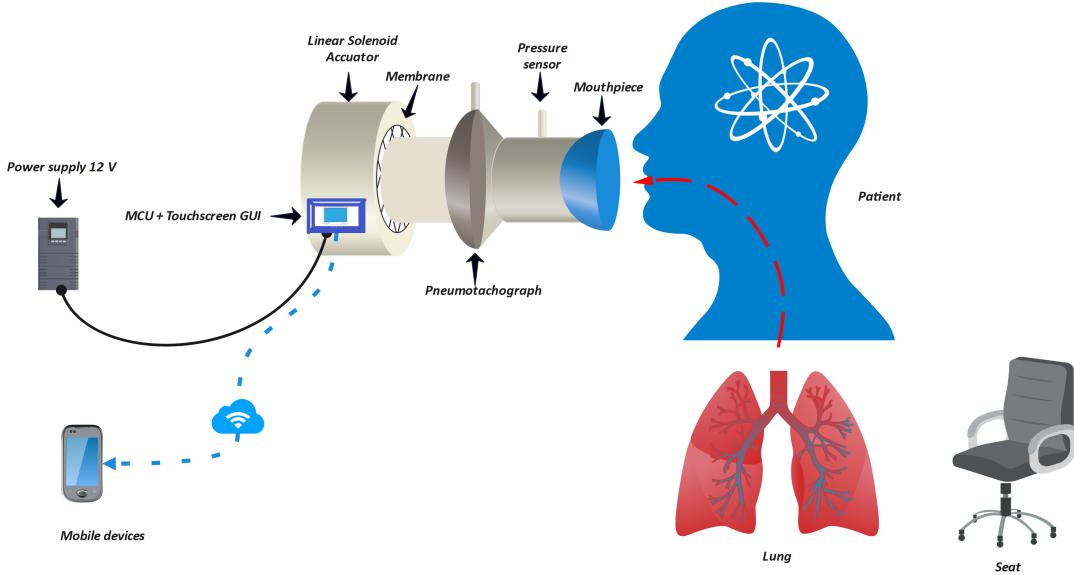


Figure 1.2: Interpretation diagram for the measurement process at phase two

sweeps from 5Hz to 30Hz after every 20 cycles of each frequency. However, this brought chaos with asynchronisation issue as the actuating signal at each frequency operated under a different time scale. Therefore, the sweeping method was changed to the time base. Each frequency sweeps after one second, which is conducive to more precise control of the **MCU**.

An interpretation diagram representing the virtual configuration of the measuring process is shown in figure 1.2, which is utilized to highlight key components required for this bio-sensing system. It is derived and improved from the preliminary design in phase one. The real configuration delivered at the end of phase two will be presented with further details in section 3.1.3.

Chapter 2

Design and Verification

2.1 Power System

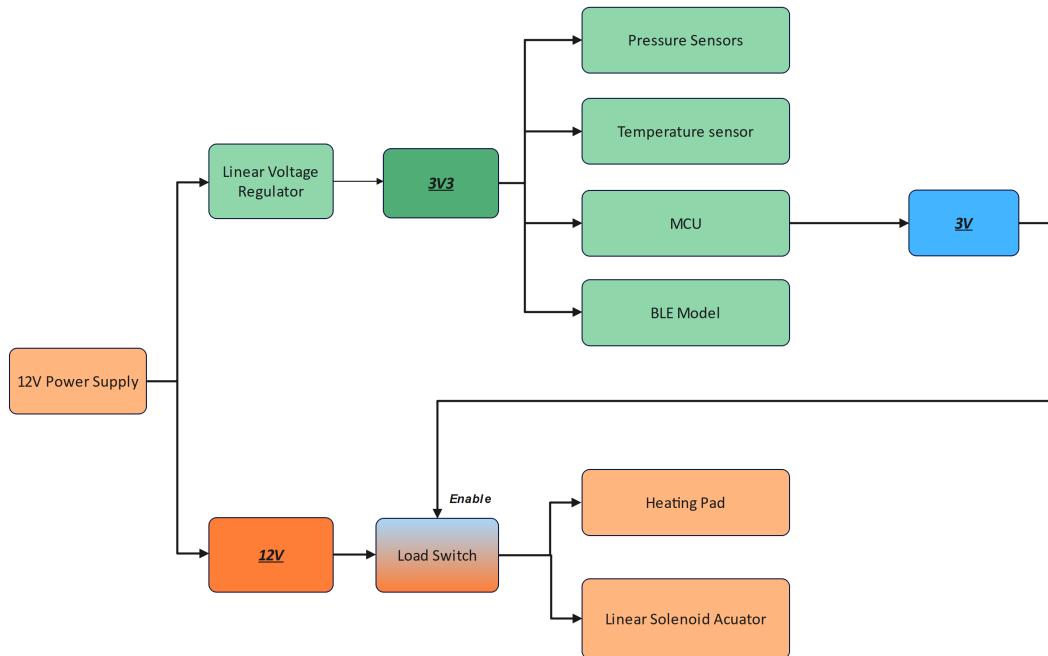
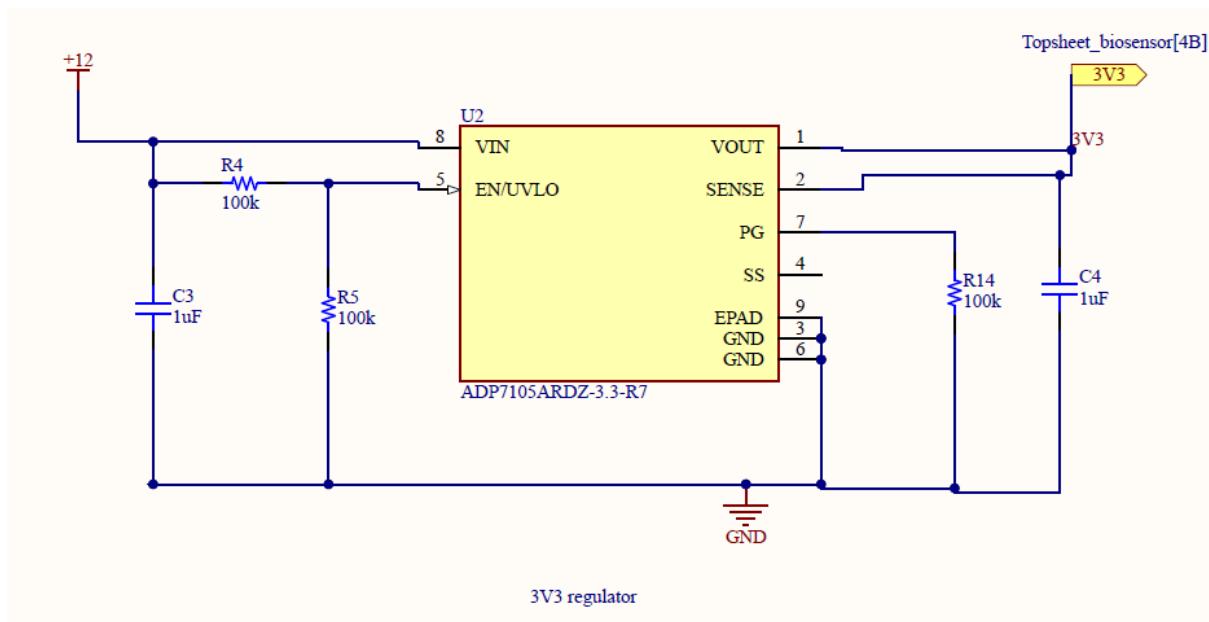
To achieve a device with low power consumption, a 12 V power supply is chosen to minimize the power conversion steps. In this case, only two voltage stages are required for the system. A power diagram of this device is shown in figure 2.1 On the one hand, sensors, the MCU and the **BLE** model are powered by 3.3 V, 12 V needs to be lowered to 3.3 V by using a linear voltage regulator circuit. On the other hand, the heating pad and linear solenoid actuator are powered directly by 12 V. However, instead of always being on, they are switched on only at specific times depending on the measuring process. An integrated load switch circuit is designed for this purpose. The enable line is powered by one of the GPIO pins of the MCU, which operates at 3 V as a logic high. Schematics and spice simulations for the 3.3 V regulator and 12 V load switch circuit are given in more detail as shown below.

2.1.1 Linear Voltage Regulator

Load current limit is the key factor that needs to be carefully considered when the designer picks up the right regulator IC. **ADP7142ARDZ-3.3**(Insert data sheet) was the original chip chosen to function as the 3.3 V regulator at phase one. However, it cannot work after experiments because of the low load current tolerance which is only up to 200 mA. The 3.3 V output is connected to the microcontroller and three sensors, which sink currents up to 250 mA and 10 mA respectively. Therefore, a new IC with part number **ADP7105ARDZ-3.3** is selected which has a load current limit of up to 500 mA. A PCB design was also updated related to this modification and summarised in section 3.1

Schematic

The schematic for the 3.3 V regulator circuit is shown in figure 2.2.

**Figure 2.1:** Power Diagram**Figure 2.2:** Schematic for 3.3 V voltage regulator

Simulation

Due to the time restriction of the PCB manufacture at the electronics lab, the verification test on the practical board was not able to be finished. However, a reliable SPICE simulation was run and results are shown in figure 2.3. The input voltage was set as a 12 V pulse wave at 1 Hz with a 50% duty circle. The load is assumed to be 11Ω to mimic the maximum operating current at around 300 mA. The graph shows the output characteristics given in figure 2.3, which verifies that the design circuit is working as it is desired.

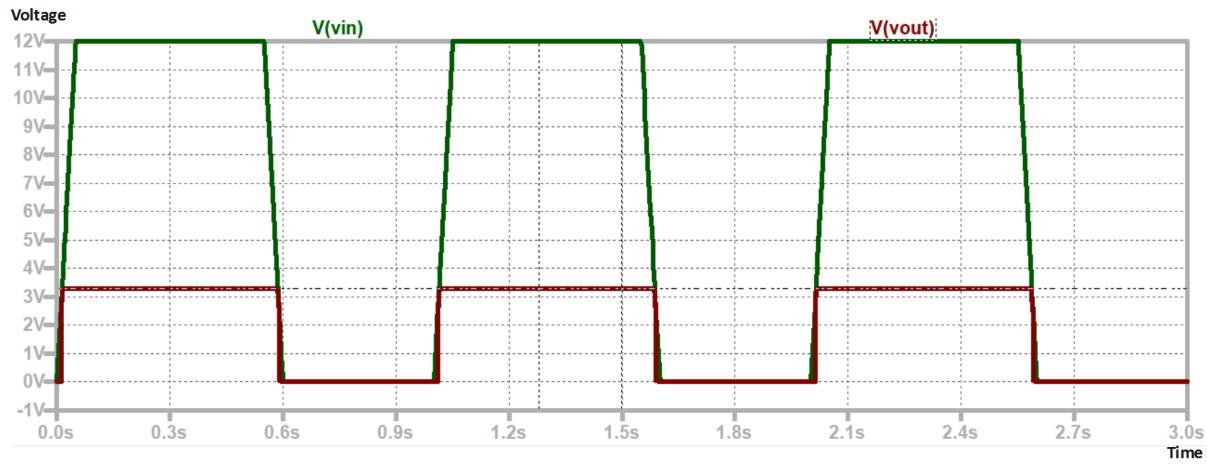


Figure 2.3: The output voltage versus the input voltage in 3.3 V regulator circuit

2.1.2 Load Switch

Instead of using a discrete MOSFET circuit, the integrated load switch circuit has many advantages in both size and feature. For example, load switches are designed into an integrated package that could even be smaller than a MOSFET itself. What's more, the load switch is embedded with many external features. **FPF2701**, as the load switch applied on this project, has an extra over-current protection circuit in the package, best suited for a large input voltage level that is up to 36 V.

Schematic

The same load switch circuits are used for switching on/off both the LSA and the heating pad. The schematic is shown in figure 2.4.

Verification

A SPICE model for **FPF2701** cannot be found in LTSpice, therefore, this part on the practical PCB was directly tested and verified. 12 V power supply was connected on board. The enable signal, as a 5 Hz square wave with 50% duty circle, was sent from MCU. Note that the **FPF2701** is low-side switched

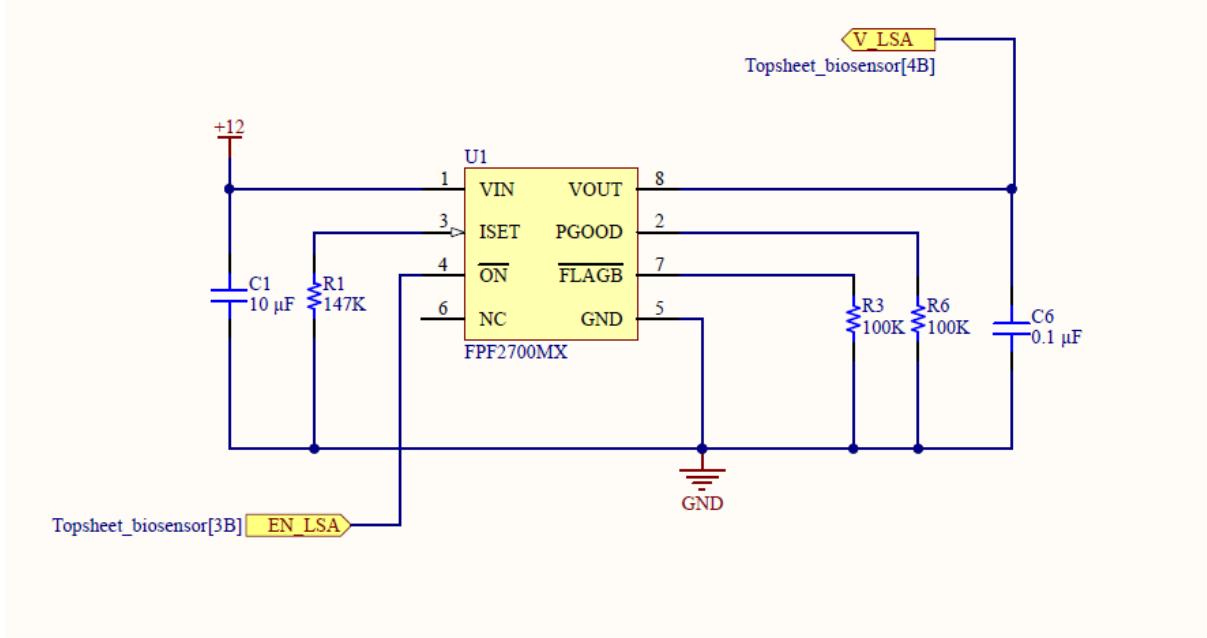


Figure 2.4: Schematics for 12V load switch of LSA

on. Results shown in figure 2.5 indicate that the output voltage varied with the ON/OFF state of the enable signal.

As it is a real circuit test, data collected from the plot is not ideal. The maximum output voltage was 11.9 V when the enable line was low, which operated as it is desired. Ideally, the *FPF2000MX* fully switches the voltage On/Off, giving the output range from 0 V to 12 V. But the minimum output voltage measured from the real verification test was 0.6 V instead of 0 V. It might be due to the manufacture restriction as only bare board can be manufactured in the university. Soldering techniques on board might be another issue, providing a poor connection between pins and tracks. However, this offset voltage level is still acceptable and could be possibly diminished after the manufacture optimization.

2.2 Sensing System

Two pressure sensors and one temperature sensor are used in this project. The general sensing process is shown in figure 2.6.

The pressure sensor with part number **SDP816-500Pa** was chosen, operating at bi-directions and giving measurement range between -500 Pa and 500 Pa. **TMP236** was selected for temperature sensing and control. All Sensors are powered by 3.3 V and have an analog output. However, when Pressure sensors are sensing continuously during the whole process, one temperature sensor is sensing discretely only for providing peripheral monitoring. For this reason, only the output of the pressure sensor is sent to a filter circuit, aiming to reduce the noise level. Then filtered/unfiltered analog signals are all directly

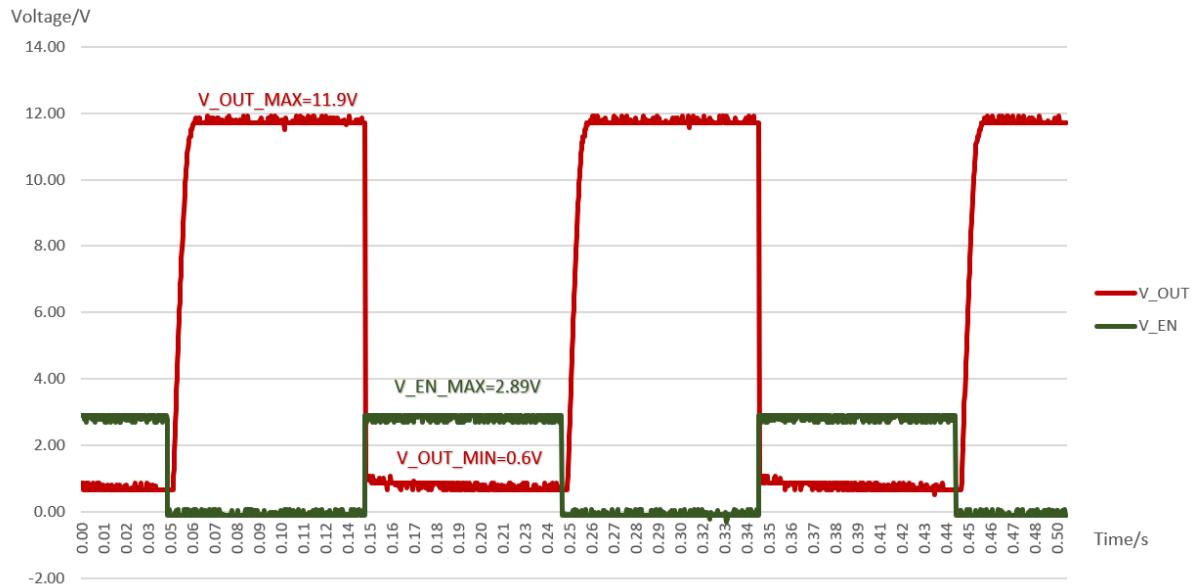


Figure 2.5: The output voltage Vs the Enable line in 3V3 regulator circuit

connected with ADCs on the MCU. The chosen MCU with part number **STM32F429I-DISC1** has three ADCs on board. The programming principle of ADC embedded on MCU will be given more details in the follow-up section 2.6.1, as a part of the control system design. Therefore, only designing of the filter circuit is illustrated next in detail.

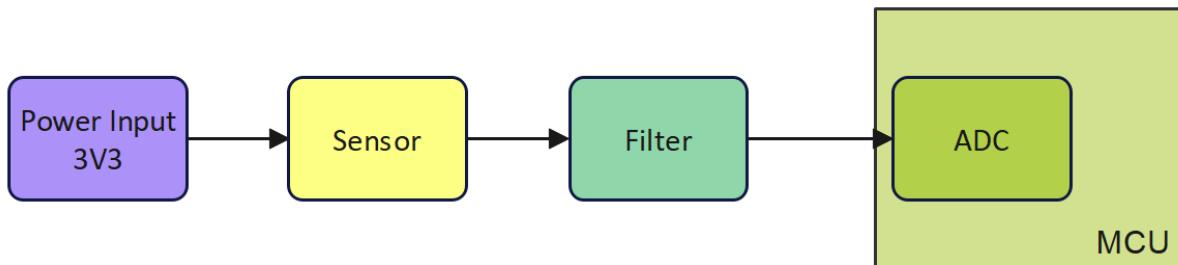


Figure 2.6: Sensing process of the device

2.2.1 Filter Designs and Simulations

Sensing signals gained from the pressure sensor could be ideally assumed as the sum of the patient's tidal breath and the actuating signal. However, The frequency range interested is between 5 Hz to 30 Hz, containing most of the useful information about small airways. Therefore, the purpose of the filter circuit is mainly to remove noise generated by the tidal breath at 0.2 Hz to 0.5 Hz.

1. Bandpass filter circuit

A passive band-pass filter was initially designed for two reasons. Firstly, due to the size restriction of the PCB and expectation of the low power consumption, the active filter circuit requires a larger space and an external power plant. A passive filter circuit was considered to be sufficient. Secondly, the initial idea was to accurately control the frequency response of the sensing signal within the range of interest. The low cut-off frequency was set to 0.9 Hz whereas the high cut-off frequency was 40 Hz.

Schematic: The schematic of the bandpass filter circuit is shown in figure 2.7.

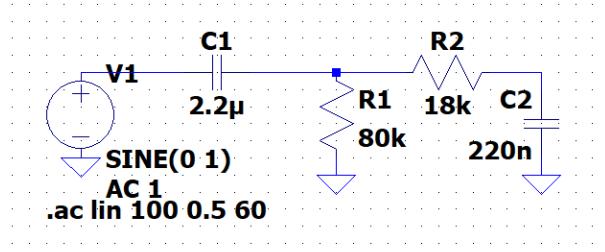


Figure 2.7: Band pass filter schematic

Simulation: The output voltage of the pressure sensor has 1.68 V offset at atmospheric pressure. Therefore, a sine wave with offset at 1.6 V, amplitude at 0.2 V, and frequency at 5 Hz were configured as the input signal to mimic a real sensing scenario. Plots were collected from both transient simulation and AC analysis of this circuit. Although the frequency response shown in figure 2.8b indicates the validation of the bandpass filter design, it still cannot be used for this device. As the negative output was detected in the transient response shown in figure 2.8a after the high pass filter part removed the offset of the input signal. ADCs on the chosen MCU cannot manage the negative input, therefore, an extra level shift circuit needs to be added.

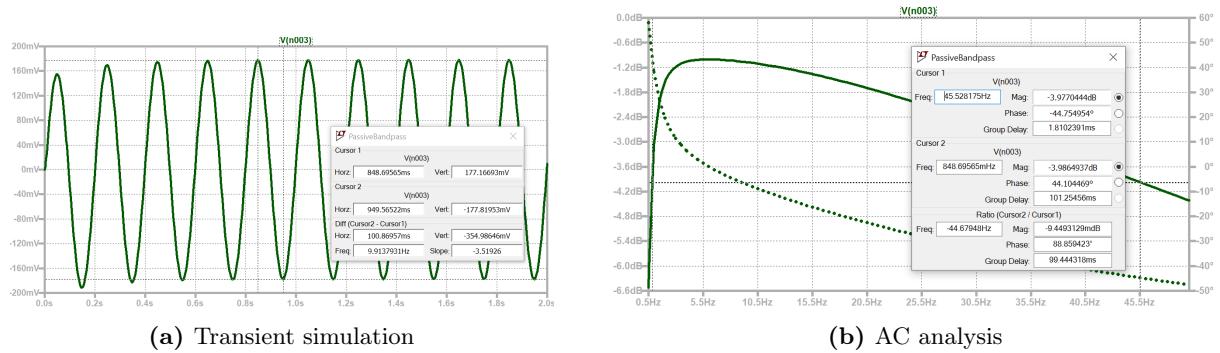


Figure 2.8: Simulation results of bandpass filter circuit

2. High pass filter Level shift

As an update, a high-pass filter with a level shift circuit took the place of the bandpass filter design. There are three main reasons. Firstly, a level shift circuit gives all positive outputs, which can be correctly read by ADCs on the MCU. Secondly, most sensing noises come from the tidal breath of the patient, which is at a low frequency. A low pass filter was considered a peripheral feature when came to the bare board design at this stage. Therefore, the low pass feature was deleted. Lastly, the passive bandpass filter also caused a -1 dB gain loss shown in figure 2.8b. By removing the low pass feature, this gain loss could be compensated.

Schematic: Schematic of the high pass filter with level shift circuit is shown in the figure 2.9.

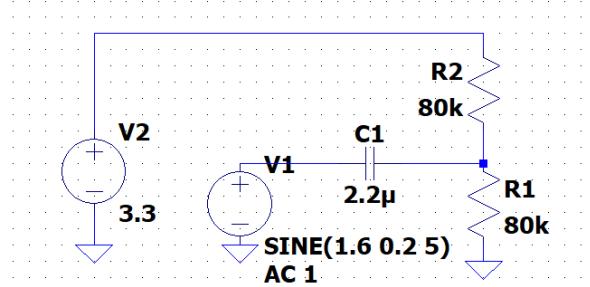


Figure 2.9: Schematic of High pass filter with a level shift

Simulation: Both transient simulation and AC analysis were run for this new filter circuit. Figure 2.10a clearly shows that the voltage level of the output is from 1.46 V to 1.84 V, shifted up by 1.67 V, which allows ADCs on the MCU to correctly read positive values all the time. The frequency response shown in figure 2.10b gives the cut-off frequency at 1.81 Hz instead of 0.9 Hz but meets the requirement. That is because a new $80\text{ k}\Omega$ resistor on the level shift part is connected in parallel with the $80\text{ k}\Omega$ resistor of the high pass filter. Therefore, overall resistance is decreased from $80\text{ k}\Omega$ to $40\text{ k}\Omega$ for the high pass filter design, which doubles the cut-off frequency according to the equation:

$$F_c = \frac{1}{2\pi RC}$$

Simulation results have proved the functionality of this filter design embedded in the device. However, due to the PCB manufacturing issue and time restrictions, real circuit tests were not finished at the end of semester one. Graphs of the patient's breath, for both tidal breath and forced breath, measured by the integrated device will be given more details in the section 3.2 later.

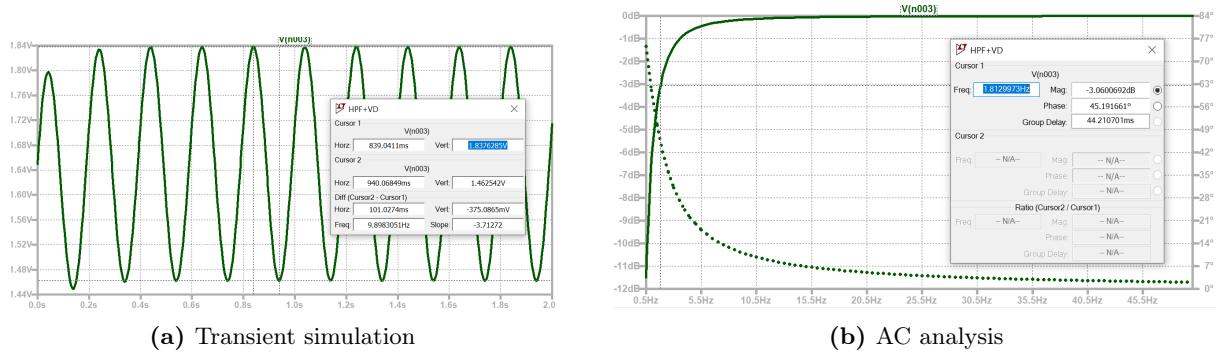


Figure 2.10: Simulation result of high pass filter circuit

2.3 Actuating System

2.3.1 Principle

As per the previous discussion in the phase one report, FOT requires an actuator to provide small-amplitude actuating signals sweeping from 5 Hz to 30 Hz. A linear solenoid actuator (**LSA**) was chosen. The principle diagram from the phase one report is given in figure 2.11.

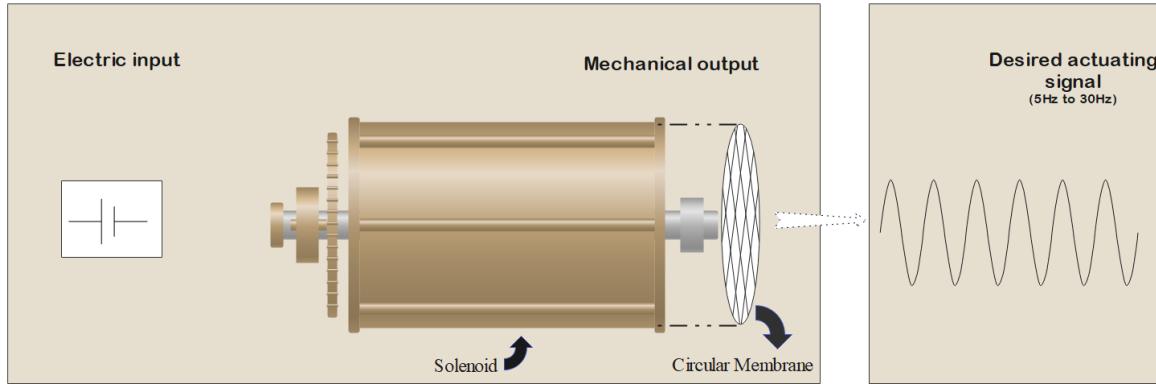


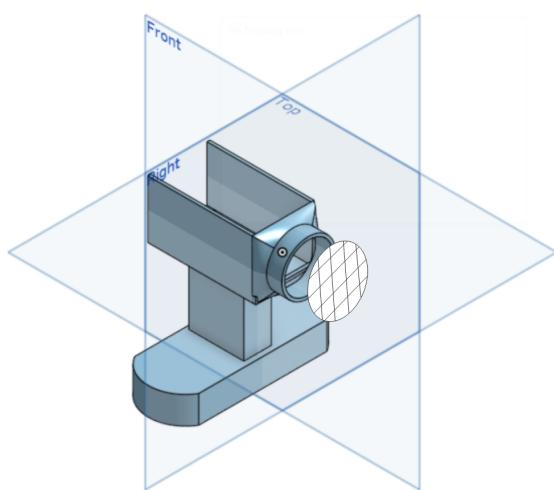
Figure 2.11: Principal diagram of LSA to generate actuating signals

There are two parts of the actuating system, a linear solenoid actuator, and a membrane. The LSA is powered by 12 V. When power is on, the piston of LSA hits the membrane, generating acoustic waves. When power is off, the spring relocates the piston, waiting for the next hit. The foundation of this process is similar to playing the drum. By sending signals from the MCU to switch ON/OFF the load switch circuit, LSA can be powered ON/OFF accordingly. The frequency of the switching signal sent by the MCU determines the natural frequency of the acoustic waves generated by LSA and the membrane.

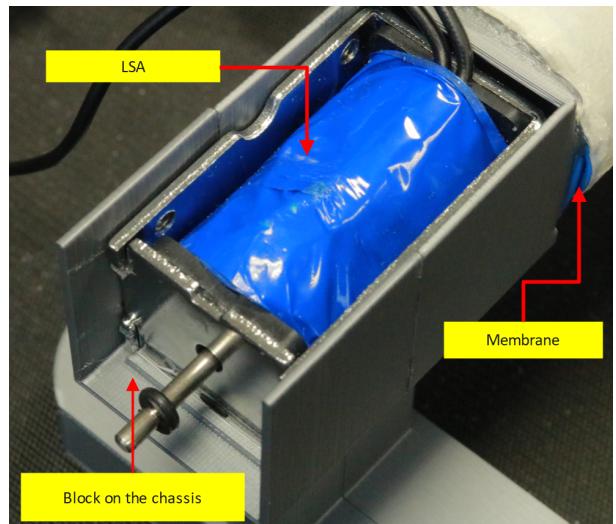
2.3.2 Configuration

There were two criteria set for choosing a LSA. Firstly, the piston has to be connected with a reliable spring, which allows the piston to be restored each time. Secondly, the piston should be well contacted with the membrane every time when it hit the membrane, which requires a relatively large force provided by the LSA and proper stroke length to fit the distance in between. To sum up, a LSA with 10 mm stroke length, powered by 12 V DC and providing 45 N as the maximum force, was chosen. Dental rubbers as the most common elastic membrane in real life were selected, which has a thickness of 0.1 mm.

A frame for the LSA was designed and 3D printed as shown in the figure 2.12a. It includes a case to fit the LSA, therefore, the dimension of it heavily depends on that of LSA itself. The more tightly the LSA fits, the less the effect of vibration would be generated during the measuring process. A block on the chassis of this LSA frame was also designed to achieve better stability, which has both a width and height of 2 mm. A membrane was attached at the circular end. By carefully setting the diameters of two junctions with a 0.4 mm difference, the membrane can be properly clamped between the LSA frame and a connector to the lung impedance sensor system. The configuration of the actuating system on the device is shown in figure 2.12b. All in all, the 3D printed LSA frame shown in figure 2.12a has the dimension in $71\text{mm} \times 90\text{mm} \times 86\text{mm}$ (*Length* \times *width* \times *height*). The detailed dimension of each 3D printing object can be found on the mechanical drawing shown in the appendix A.¹



(a) Selected LSA



(b) The configuration of the actuating system on device

Figure 2.12: Actuating System

¹STL files of all 3D printing objects can be found in the folder named *MechanicalOutput* through GITHUB link <https://github.com/Peggy-p0799/Biosensor-UI-MCU>.

2.3.3 Verification

Methodology and Result: Real actuating signals generated by the LSA were measured after the final integration of the device. All data was collected directly from the same pressure sensor and measured by an oscilloscope. Figure 2.13a indicates what actuating signals look like during the whole measuring process. It is clear that the frequency of the actuating signal is increased between 5 s and 30 s, which exactly meets the original requirement and expectation.

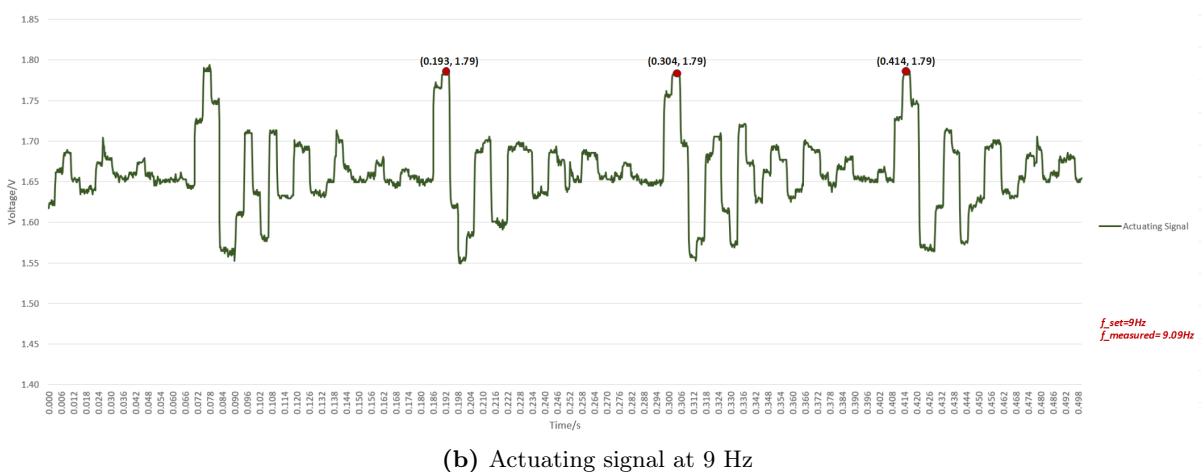
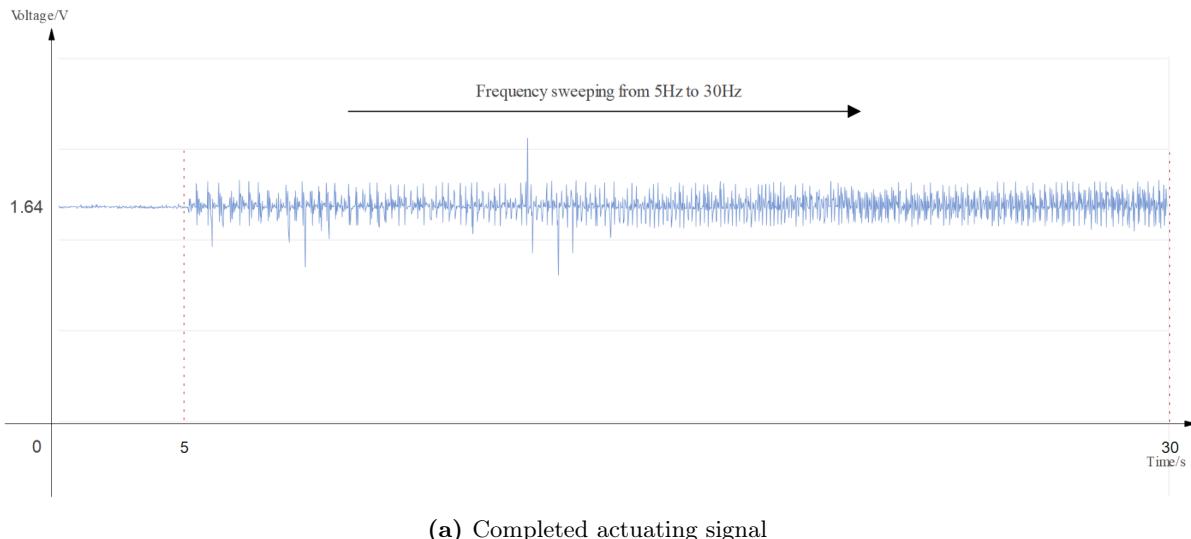


Figure 2.13: Components selected for the actuating system

For further analysis, a fragment of this waveform is selected, amplified, and analyzed, which is shown in figure 2.13b. Two pieces of main information carried out by this plot are used to verify the reliability of the actuating system. Firstly, this actuating signal is a sinusoidal waveform with a single natural

frequency and multiple harmonic frequencies, which matches the theory about the vibration of a circular membrane discussed in the phase one report. Secondly, the natural frequency of it was calculated from two different peaks, which is 9.09 Hz, around 9 Hz, consistent with the integral actuating frequency originally configured between 5 Hz to 30 Hz. Therefore, it is an approachable method to generate actuating signals required for FOT.

Limitation: However, there are also many unknowns and limitations of this actuating system.

For example, the 1st harmonic frequency should be calculated and analyzed for each signal under a different natural frequency. If the harmonic frequency is smaller than 30 Hz, it may affect the accuracy of the actuating process as well as the final data captured from the forced breath.

What's more, the amplitude of the actuating signal has to be clarified by doing more experiments on the properties of the membrane. The FOT requires a small amplitude actuating signal. However, a forced breath was measured with actuating signals applied as shown in figure 2.14. It seemed that this actuating signal was too big to be within the sensing range. Inappropriate actuating signals cause the distortion of the patient's breath, which may be the reason why expected impedance cannot be attained at the end of phase two. The limitation of the current actuating system and sensing system will be discussed again in section 3.2 to analyze the undesired measurement result.

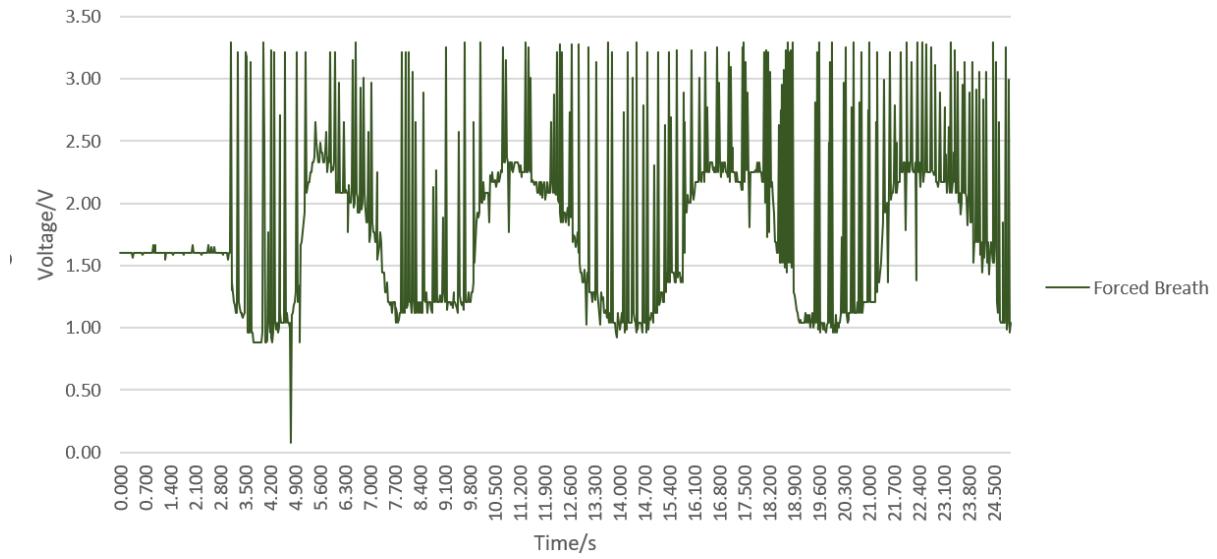


Figure 2.14: Forced breath with actuating signals applied

2.4 Heating System

2.4.1 Principle

Generally speaking, there is around 15 °C difference between a human's mouth(37 °C) and room temperature (22 °C). When patients use this device at room temperature, vapor condensing on the sensing part causes major inaccuracy in the final measuring result. Therefore, a heating system was set up in order to minimise the impact of temperature differences. The block diagram of the temperature control system is shown in figure 2.15.

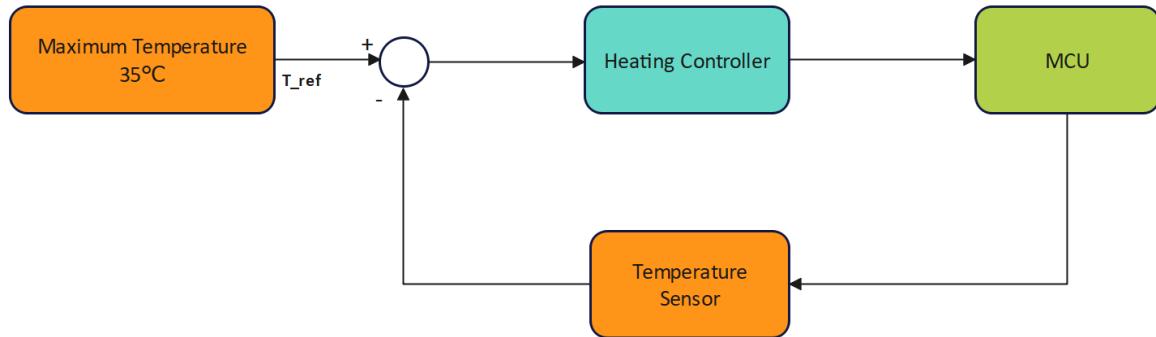


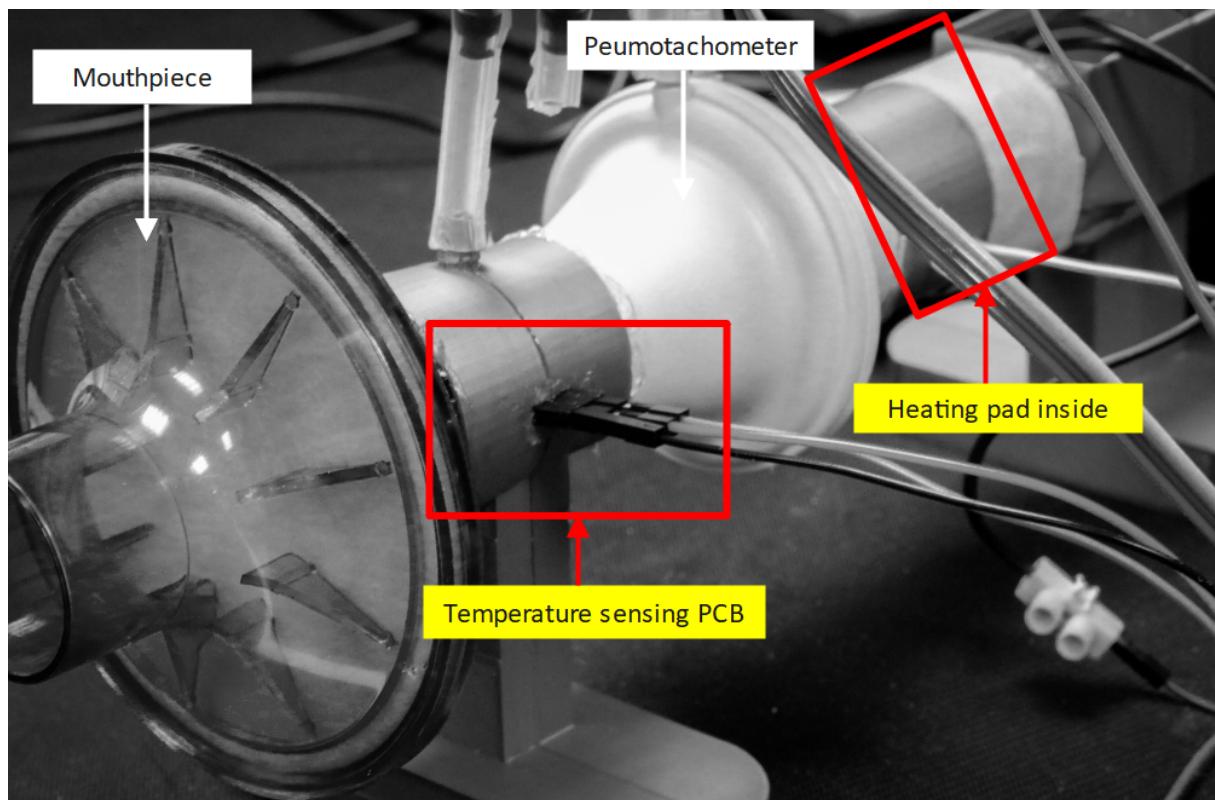
Figure 2.15: Block diagram of the temperature control system

Single-stage control with only one temperature sensor was designed initially. Temperature reference was set to 35 °C that is approximately equivalent to the temperature of the air out of the human body. A heating pad was stuck on the entrance of the air outlet of the device, heating up the air under room temperature. When the sensor detected temperature was below 35 °C, MCU controlled the enable line of the load switch circuit to logic low, the heating pad was powered and started heating. Once the temperature captured was greater than the temperature reference, the heating element was switched off by the MCU. The output of the temperature sensor is analog, connected with the ADC on MCU. This ADC function was switched ON/OFF every 500 ms, therefore, the state of the heating pad was also checked every 500 ms.

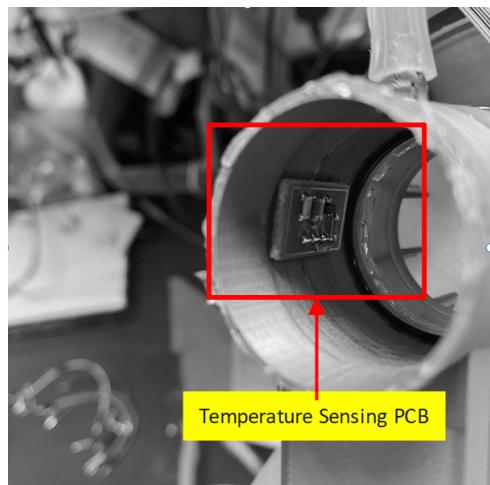
2.4.2 Configuration

The Configuration of the heating system on the device is shown in figure 2.16a. There are two facts when comes to the application of this heating system.

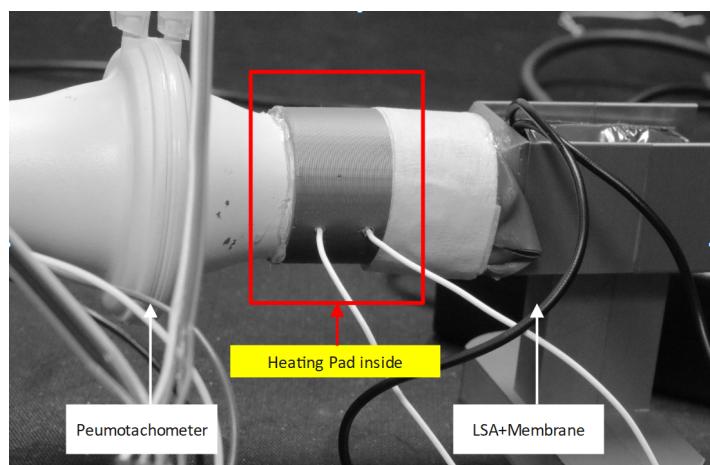
- A small PCB board was designed for *TMP236* sensor due to a peripheral circuit required and assembly condition related. The size was restricted by the diameter and the length of a connector between the mouthpiece and the pneumotachometer. The board is attached to the wall of the



(a) Configuration of the heating system on device



(b) Temperature sensing PCB on device



(c) Heating pad on device

Figure 2.16: Heating system

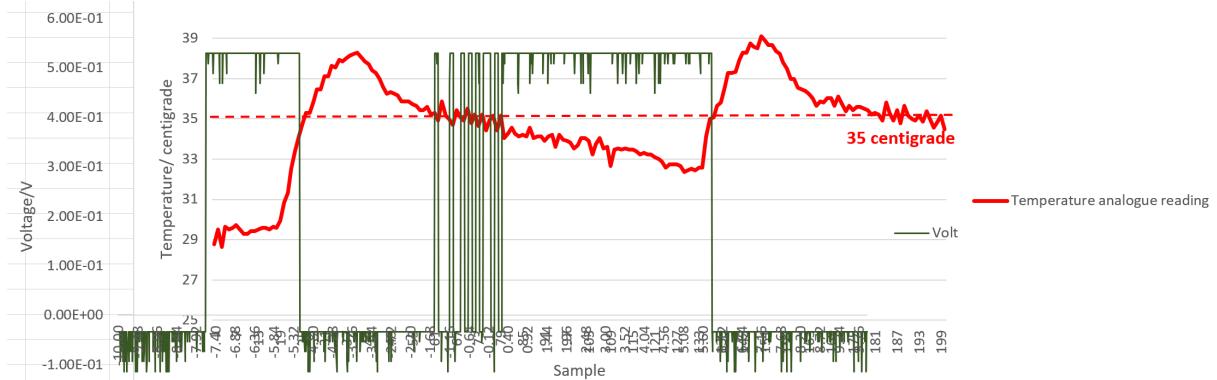


Figure 2.17: The overlapping plot for I/O of a temperature control process

pneumotachometer input as shown in figure 2.16b. More information about this PCB layout is given in the section.

- The heating pad was configured at the entrance of the air outlet as shown in figure 2.16c, which is at the output of the pneumotachometer.

2.4.3 Verification

Methodology and Result: The process for a verification test of the temperature control logic follows the steps given below:

- Connect the temperature sensing PCB with the MCU, and power both of them properly.
- Start the temperature control function on MCU. Hold the sensing PCB inside two hands, and breathe through the only small hole reserved by hands.
- Breath in 2 seconds, stay away and let it cool down for 5s, repeat this process in two circles.
- Collect data from both the analog reading of the temperature sensor and enable signal sent out accordingly on MCU.
- Overlap two graphs to see if the control process is reliable.

The overlapping for the analog input and digital output is shown in figure 2.17. It proves that the current heating system provides a reliable response regarding the changing temperature.

Limitation: The heating system was a peripheral function of this device, insufficient consideration has led to many problems with this system.

For example, the verification test mentioned above is uncompleted. It can only verify the reliability of the control response, however, results showing whether the temperature can be exactly kept at 35 °C was not verified. This is because of the lack of another sensor. A new scenario must be designed to make sure

that no matter how the temperature changes at the entrance, the temperature inside the instrument is always maintained at 35 °C.

In addition, the distance between the temperature-sensing PCB and the heating pad was too big. Without another sensor on the heating pad, the attached shell might be overheated due to this configuration. At least two sensors will be required if accurate control is expected.

Finally, the current control logic was too simple, which only switches the heating pad ON/OFF accordingly. There is no PID controller to precisely calculate how much temperature needs to be compensated. The temperature control system can be far more advanced with lots of improvements after further consideration.

2.5 Communication System

As a device design, a user interface shall be embedded for users to talk to the device. Therefore, the communication system contains two parts, a touch screen and a BLE module, which allows users to transmit instructions as well as receive data.

2.5.1 GUI- TouchGFX

The GUI, known as the graphic user interface, is a form of that allows users to interact with electronics through graphic icons and audio indicators.[2] A touchscreen GUI is so intuitive that users can simply manipulate it by fingertips or with a stylus. **STm32F429i-Disc1**, as the chosen MCU, has a resistive touch screen with **STMPE811** controller onboard². All components are from STMicroelectronics family, which makes the programming process and integration fairly easy.

Principle

Software: The development of the communication system relies on two software, a graphic C++ tool TouchGFX and an advanced C/C++ development platform STM32CubeIDE. The TouchGFX provides a quick and intuitive solution to design a GUI framework for an STM32 microcontroller. Besides, It also supports STM32 files as the I/O, and the designed framework can be directly converted to an STM32 project and opened in STM32CubeIDE for further optimisation. Therefore, the overall structure of GUI was first designed in TouchGFX. Functions as the interaction with the MCU were then written in detail in STM32CubeIDE.

Framework: GUI framework was designed as shown in figure 2.18. The touchscreen has 11 pages in total. Before starting a new measurement, the instruction for the device is first provided. User needs to carefully read this information and hit the “AGREE & CONTINUE” button on the next page. Then personal information is collected as respiratory impedance is closely related to the physiological information of the patient. Then, when the user toggles on the start button, the function of activating

²More details of the STMPE811 can be found at <https://stm32f4-discovery.net/2014/05/library-10-stmpe811-touch-screen-driver-for-stm32f429-discovery-board/>

a new measurement is called. During the measuring process, a progress bar is constantly displayed, revealing the progression. The measurement usually lasts 25 s, with each frequency swept every 1 second. After the measurement gets finished, the user can choose to whether check plots of the respiratory reactance and resistance separately or directly save all numerical data through the BLE module.

Architectural Pattern of GUI - MVP: TouchGFX GUI design follows an architectural pattern called Model-View-Presenter, derived from another pattern known as the Model-View-Control. Both of them are commonly used for the development of user interfaces. Three classes in MVP are defined below:

- The *model* is a repository for data and actions that is to be executed. It is an interface with a UI system and non-UI backend system, the MCU in this case.
- The *view* is the visual layer of the GUI. All data that needs to be shown in the screen is finally sent and called here.
- The *presenter* is the logic layer of the GUI. It retrieves data from the model and formats it for display in the view.

A diagram from an official document [3] provided by TouchGFX illustrates the relationship between these three classes and external communication, which is shown in the figure 2.19. For a further explanation, a progress bar design is given in figure 2.20 and discussed below in detail as an example of the communication system design.

The progress bar is one of the touchscreen outputs as discussed in the framework item above. It shows the real-time progress of the measurement. Therefore, the percentage of progress displayed on the screen needs to be dynamically updated from the MCU. *Task 2: Frequency Sweeping* is One of the freeRTOS tasks generating actuating signals and sweeping its frequency from 5 Hz to 30 Hz. A queue is used to transmit and receive data from different tasks and systems. So the real-time frequency is transmitted by a queue from the MCU and received by the model of GUI. The logic is developed in the presenter to convert the frequency to the percentage of progress. Finally, the formatted percentage data is sent to the view layer and displayed on the touchscreen.

Although progress bar design is the only example given details here, all other functions, like triggering a new measurement, plotting the impedance, and saving all data, that interact with the MCU, are all designed based on the MVP pattern, following a similar process. Because the MVP pattern of GUI divides codes into several parts, making them more readable, usable, and maintainable. It provides high flexibility to manage functions and debug for a GUI development that should be carefully considered and implemented.

Verification

As a user has to participate in the verification of GUI design, there is no specific data or screenshots to prove its functionality. However, a short video named as **operationRecording.mp4** was recorded by

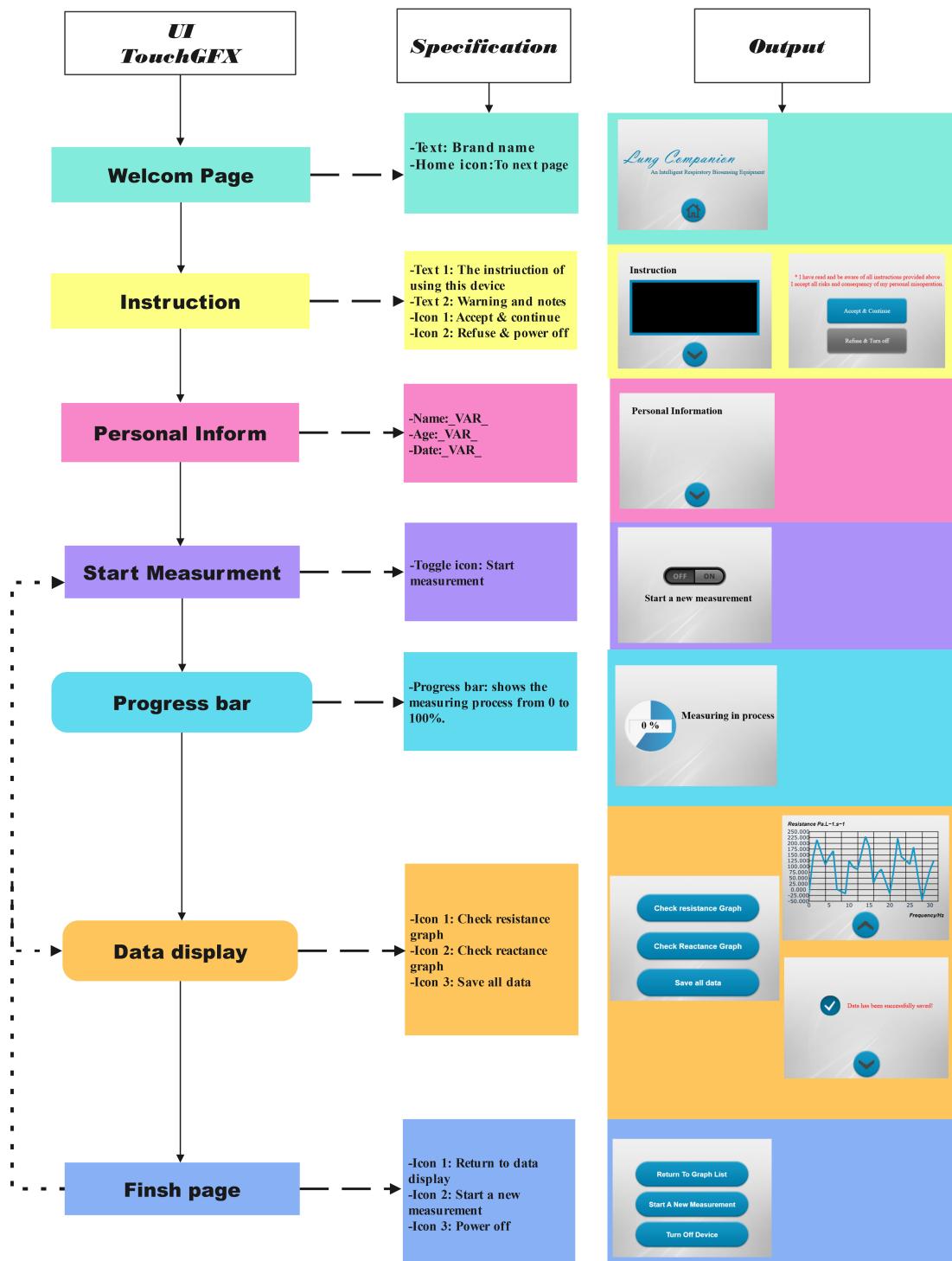


Figure 2.18: GUI framework and output

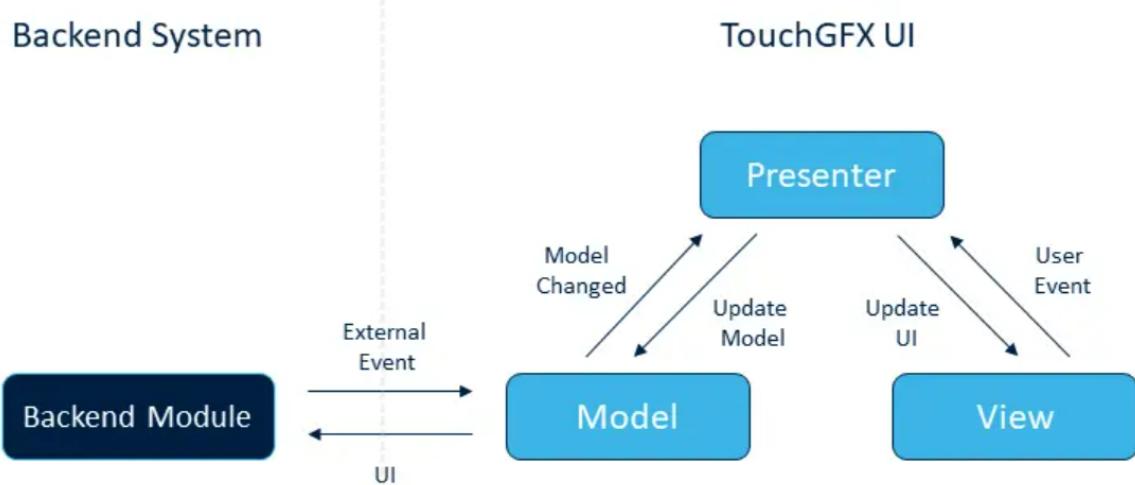


Figure 2.19: Model-View-Presenter and external communication [3]

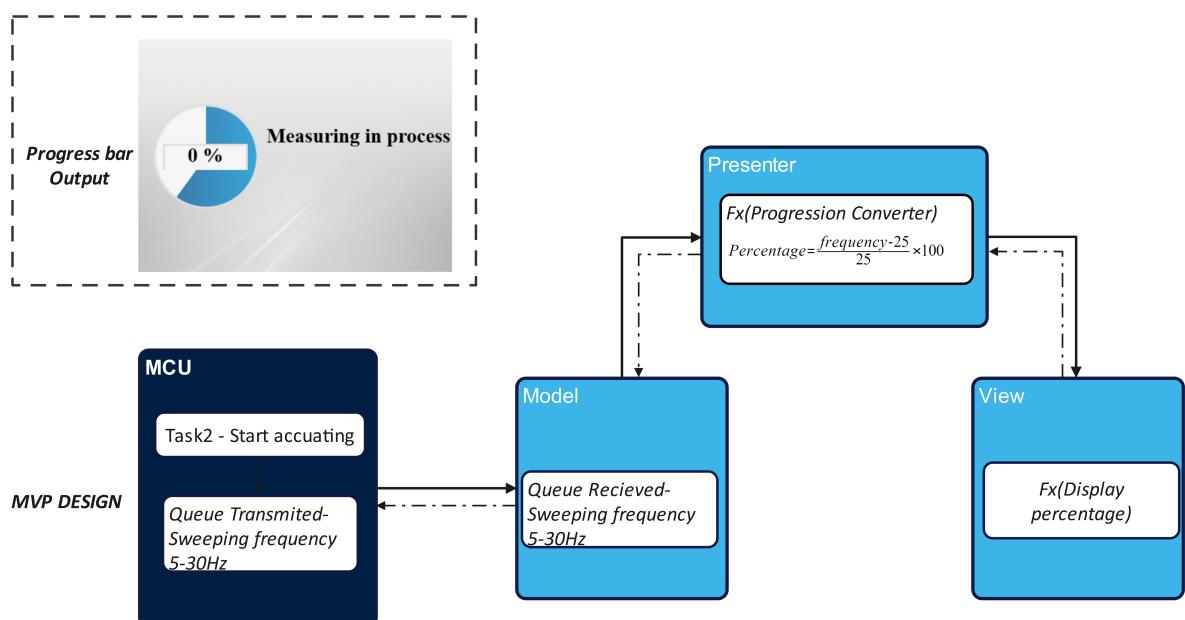


Figure 2.20: Progress bar example - MVP design

the project student, which records the complete working process of the current device ³, proving that all GUI design has been put into practice and functions properly.

2.5.2 BLE model

Principle

BLE, known as Bluetooth Low Energy, is a wireless communication technology, designed specifically for a low-power consumption application. It is a peripheral functionality that was expected to be applied to this lung measurement device as a part of the communication system.

There are a few criteria to successfully pick up a BLE module. Firstly, the compatibility of BLE and interfacing MCU should be carefully considered. Some BLE modules only support a specific protocol to interface with the microcontroller. For example, a BLE module only supports I2C protocol and cannot interface with a MCU that is not equipped with I2C ports. Besides, the version of the BLE module is also very important, which limits the device that can interact. For instance, the IOS device only supports BLE 4.0 module, whereas most Android devices support both BLE 2.0 and 4.0 modules. Lastly, different BLE modules offer different features such as speed, throughput, and operational range. Therefore, specific selection needs to coincide with the demand of different applications.

The HM-10 is a Bluetooth 4.0 module that was selected for this application. It is based on the Texas Instruments *CC2540* or *CC2541* BLE System SOC (*System on Chip*) [4]. Its firmware is managed and manufactured by a company called DSD tech [5]. The HM-10 stands out for its high data rate of up to 24 Mbps, long-distance communication of up to 100 m, simple serial UART connection, and the capability to interface with the IOS device of the developer. A circuit diagram illustrating the configuration between HM-10, MCU and mobile is shown in the figure2.21. More details for programming BLE through UART protocol on MCU will be given in the control system design section 2.6.1.

Verification

Methodology: There are two purposes for operating the verification test of BLE module. Firstly, the test should verify that measured data can be transferred from MCU to a BLE app on mobile. Secondly, the test should verify that correct data is transferred from MCU to a BLE app on mobile.

Related code on board has been verified to work properly in debugging stage. The introduction text was added before transferring numerical data on respiratory impedance. The test has the following process:

- Connect HM-10 with MCU as the configuration shown in the figure 2.21. Make sure HM-10 is powered by 3V and a red LED on the board starts blinking.
- Connect HM-10 with a mobile through an app called *BLE SCANNER*. Check the state of the red LED on HM-10. The connection has been set up if it stops blinking
- Run the code, observe and record the read output on APP.

³The operation video can be checked through the GitHub link: <https://github.com/Peggy-p0799/Biosensor-UI-MCU>

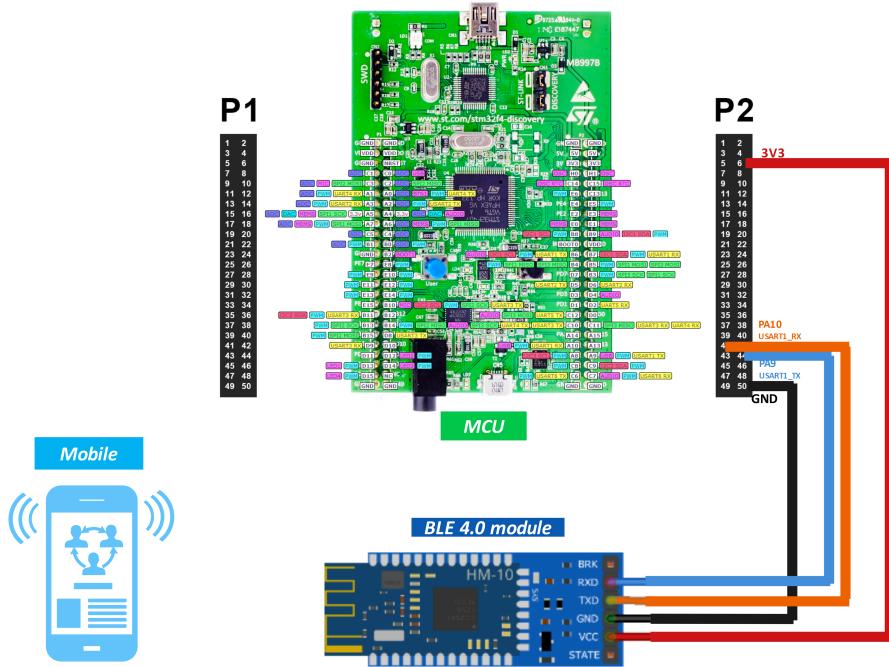


Figure 2.21: BLE connection diagram

Result: The code and screenshot of the reading on the APP are combined and shown in figure 2.22. It indicated that the BLE module was working and data kept exchanging through UART lines between HM-10 and MCU. However, only three ASCIIIs were received each time. This was not desired as UART on MCU was used to transfer an 8-bit string array at once. It might be caused by the rate mismatch between HM-10 and UART rate configuration on MCU. Alternatively, it might directly be limited by inherent reading configuration on the APP. As the BLE module was designed as a peripheral function and time was restricted at the end of semester two, wireless communication on board was not amended and verified properly. The reliability of the BLE module still requires lots of effort at a further stage.

2.6 Central Control System

2.6.1 Principle

The control system is the core of this biosensor device. It delivers five complicated functions that are collecting data from sensors, calculating the respiratory impedance, sending out actuating signals, communicating with GUI and BLE modules as well as controlling the temperature inside of the device. *STM32F429I-DICS1* discovery board is from STMicroelectronics, chosen here as the MCU. The system design was developed on MCU by an advance *C/C++* programming platform called *STM32CubeIDE*.

There are 5 functions designed in the control system, and all of them have to be processed at the same



Figure 2.22: Verification data collected through BLE module

time when measurement starts. However, a normal control system only allows one task to operate at each time. Therefore, the FreeRTOS (*Real-time operating system*) was considered to be used for solving the single-threaded issue. As a real-time operating system kernel for embedded devices, freeRTOS is perfect for multiple threads, tasks, timers, and so on. Luckily, it is a popular tool and supports over 35 platforms which include ARM Cortex-M that is embedded on *STM32F429i-DISC1*. The current control system developed on this MCU depends fully on FreeRTOS, individual tasks were set for each function and would be explicitly explained below.

FreeRTOS

In freeRTOS, a real-time application can be structured as a set of different tasks. Every task is independent of any other task within the system [6]. By setting the priority and stack size of different tasks, multi-threads can execute in one operating system. But how to synchronize different tasks and operate them at specific timing? FreeRTOS offers a lot of kernel objects for thread-safe communications between threads such as queues, mutexes, and semaphores. Queues, as the most basic and efficient object, pass information without interrupting tasks, which were used for task synchronization and data transfer in this control system. Both queue size and item type can be customised for each queue. A list of tasks and queues created for the biosensor system is shown in figure 2.23

There are five tasks and five queues in total. A general introduction is firstly summarised below, more details of each task will be given in the next subsection 2.6.1.

Tasks

- **GUI-Task:** It is the GUI task that is generated by *TouchGFX*. It shall be called once the

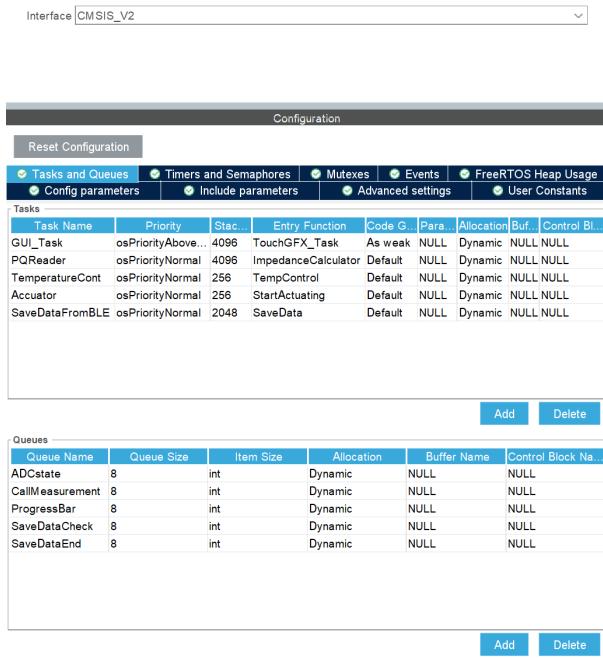


Figure 2.23: Configuration of Task and Queue in freeRTOS

device is turned on. Therefore, the priority of it was set to the highest (above normal). 4096 words were given as a relatively large stack size for complex images and functions of GUI.

- **PQReader:** This task is for respiratory impedance calculation. It collects sensing data from ADCs on board and does an FFT(*Fast Fourier Transform*) to get respiratory resistance and reactance separately. As results are expected as precise as possible, 2048-points FFT was chosen to process data at each specific frequency. Therefore, the stack size of this task was set to 4096 words.
- **Accuator:** This is a task for generating actuating signals to switch On/Off the LSA actuator when a new measurement starts. It would only be called once the user hits the *start measurement* toggle button on the touchscreen. Therefore, it has a lower priority compared to GUI Task, but the same priority as *PQReader*. Not much data is dynamically transferred inside this task. The stack size of it was set to 258 words.
- **TemperatureCont:** This task is for a single-stage temperature control. ADCs onboard reads data from a temperature sensor and send out logic High/Low enable signal through a GPIO pin to switch the heating pad On/Off accordingly. Similarly, it has a normal priority as it starts only when a new measurement begins. A small stack size was also given to this task for better memory management.
- **SaveDataFromBLE:** This task is designed for MCU to communicate with BLE module through UART protocol. It will only be called once the measurement is finished and the user hits the *Save All Data* button. The amount of data flow in this task and its stack size was set

to 2048 words.

Queue

- **ADCstate** A queue passes the information on whether all data has been converted by ADCs at each working frequency. ADCs on board were designed to collaborate with the DMA(*directly memory access*), for effectively dealing with the amount of data. DMA buffer size was designed according to the sampling rate of ADCs so that the buffer can be exactly filled before frequency sweeps to the next. More details about the relationship between the sampling rate of ADCs and DMA buffer size are given in section 2.6.1. When the buffer is full, *ADCstate* changes from 0 to 1, which is sent to *Task: PQReader* and *Task: Actuator* to synchronize sensor readings and frequency sweeping of actuating signals.
- **CallMeasurement** A queue passes information whether *Start a new measurement* toggle button is clicked on touchscreen GUI. When the user toggles on this button, *CallMeasurement* changes from 0 to 1 to start threads related to the measurement.
- **ProgressBar** A queue to pass the current actuating frequency from *Task: Start Actuating* to *Task: GUI-Task* so that progress can be converted and displayed on the touchscreen.
- **SaveDataCheck** A queue passes information whether *Save All Data* button is clicked on touchscreen GUI. When user clicks this button, *SaveDataCheck* changes from 0 to 1 to start *Task: SaveDataFromBLE*
- **SaveDataEnd** A queue passes the information on whether all data has been successfully saved. This queue will send 1 to the touchscreen GUI, indicating that the data transfer is completed and the screen shall be changed to the next.

The purpose of Queues is mainly for synchronizing tasks. Computed impedance and measured sensing data are not transferred by any queue. Therefore, 8 words *int* was configured as the type of passing item for all queues.

Task 1 - Impedance Calculation

Sampling rate: According to the Nyquist Theorem, the sampling rate of ADC(f_s) needs to be at least twice the maximum frequency of the converted signal to recreate the original analog signal reliably. In this case, f_{max} is 30 Hz. Therefore, the minimum f_s shall be 60 Hz. However, the accuracy of sensing data is expected as high as possible, sampling frequency shall be given at around KHz level. By considering the fact that the actuating signal sweeps its frequency from 5 Hz to 30Hz every second, the sampling frequency was set at 2.048 kHz and the DMA buffer size was set as 2048, which makes sure the sensing data of P (pressure) and Q (flow rate) have been collected and transferred for impedance calculation at the end of each actuating frequency.

Note that ADCs onboard can be triggered by any of TM1-5 or TIM8 to synchronise A/D conversions and timers. Therefore, the sampling frequency of ADC was configured on the timer panel, TIM3 in this case, instead of the ADC panel. Detailed configurations for ADCs and timers on *STM32CubeIDE* be

Listing 2.1: Analogue conversion code in Task 1

```

fftin_buf_p[n]=((float32_t)adc_buf_p[n]*3/4096/3.3-0.5)*pow(((float32_t)adc_buf_p[n]
*3/4096/1.32-1.25),2)*525; //Pressure sensor conversion (Pa)

fftin_buf_q[n]=(((float32_t)adc_buf_q[n]*3/4096/3.3-0.5)*pow(((float32_t)adc_buf_p[n]
*3/4096/1.32-1.25),2)*525)*2.083/98.066; //Flow rate conversion (L/s)
\label{lis:conversionTask1}

```

checked through the real STM32 project uploaded on GitHub [7]. More information about the pinout view and system view on *STM32CubeIDE* can be found in appendix C.

Analogue conversion: Before getting the frequency response of sensing data, it needs to be converted from digital reading back to analog reading. 12-bit ADC was configured to provide a digital reading of sensing data within the range $0 \sim 4095$ ($2^{12} - 1$). Therefore, the relationship between analog reading and digital one is based on the equation:

$$A_{reading} = \frac{V_{ref}}{2^{12}} \times D_{reading} \quad (2.1)$$

where V_{ref} is 3 V. Conversion to physical values for pressure sensor(*SDP816-500Pa*)[8] and pneumotachometer (*MLT300L*)[9] are given on their data sheet respectively. *MLT300L* is assumed to be linear between $0 \sim 300$ L/min. Equations were calculated and applied to the code of task one as shown in listing 2.1.

The unit of pressure was unified as Pa and that of flow rate was unified as $L s^{-1}$, which gives $Pa \cdot L^{-1} \cdot s$ as the unit of computed impedance.

FFT on STM32: The principle of the impedance calculation has been explicitly illustrated in the phase one report. The FFT(*Fast Fourier transform*) is a vital tool for frequency analysis. Another function called *DoFFT* was written to apply the fast Fourier transform on the sensing data and compute respiratory resistance and reactance at each actuating frequency accordingly. The CMSIS DSP(*Digital Signal Process*) library provides FFT functions, which can only be implemented on STM32F4 and STM32F7 series. This was the pivotal reason why the *STM32F429I-DISC1* discovery board was finally selected as MCU for this device.

To successfully apply FFT on the STM32F4 series, two tips are noted here.

1. Select a proper FFT length N . There are few rules to limit the choice. Firstly, sample counts N have to be the power of 2 from 2^4 to 2^{12} . Secondly, one FFT process gives two outputs, the real part, and the imaginary part. For this reason, the size of the input buffer has to be two times the FFT size ($2N$) to get a complete frequency response. Output buffer size is the same as N . Lastly, to locate the desired frequency response at the output of FFT, the frequency resolution is often

calculated depending on the sampling frequency and the FFT length as shown in equation (2.2).

$$\text{Resolution} = \frac{f_s}{N} \quad (2.2)$$

In this case, f_s is 2.048kHz. Because the input of FFT is the output of ADC, the sampling frequency of FFT is consistent with that of ADC as configured before. Frequency resolution was set to 1Hz as actuating frequency swept from 5 to 30Hz, increased by 1Hz at each time. Above all, FFT length N was configured as 2048, as same as the input and output buffer length. Although it did not fit the second rule mentioned above, however, the interesting frequency is from 5~30Hz, and only the first 60 outputs are interested. 2048 is acceptable as the FFT length for this design.

2. Include the *arm_math.h* head file and CMSIS DSP library in the workspace.

Task 2 - Frequency Sweeping

PWM configuration: As mentioned above 2.3.1, LSA is controlled by a load switch circuit. By sending an enable signal from MCU to the load switch circuit, LSA can be switched On/Off accordingly. The purpose of this task is to generate an actuating signal sweeping its frequency from 5Hz to 30Hz.

The PWM(*Pulse Width Modulation*) method was utilised on MCU to send out this enable signal. It is usually used for DC motor control. By switching the pulse frequency up to kHz, discrete switching will not affect the load supply, instead, the average-voltage level can be simply varied by changing the duty circle of the pulse. However, it can also be a communication method, delivering information over different interfaces. In this case, frequency sweeping of the enable signal can be easily achieved by dynamically changing the frequency of PWM.

The PWM is generated by the timer on STM32 microcontrollers. The configuration of PWM onboard shall revolve around its two main components, duty circle, and frequency.

- **Duty circle:** the ratio of the time of **High** state and the total time in one cycle. The duty circle of PWM depends on ARR(*Auto Reload Register*) and CCR(*Capture/compare Register*) on STM32s. The relationship between the three of them is provided in an application note of *STMicroelectronics* about STM32 cross-series timer [10], which is shown in the equation 2.3.

$$\text{Duty\%} = \frac{\text{CCRx}}{\text{ARR}} \% \quad (2.3)$$

- **Frequency:** how fast the PWM can be switched On and Off. There are two parameters that define the frequency of PWM, ARR, and pre-scalar of the APB real-time clock. The relationship between the three of them is provided in the same document by *STMicroelectronics* [10] and summarised as shown in the following equation 2.4.

$$\text{Frequency} = \frac{\text{APB_TIM_CLOCK}}{(\text{PRESCALAR} + 1) \times (\text{ARR} + 1)} \quad (2.4)$$

In this case, LSA should be switched On and Off within the same amount of time. Therefore, the duty

circle was set to 50%. The PWM signal aims to achieve frequency sweeping from 5 to 30Hz. As ARR and CCR are register value that has an impact on the resolution of PWM and the constant duty circle, respectively. Therefore, frequency sweeping shall be processed by changing the value of the pre-scalar instead. The APB real-time clock was set to 72MHz for MCU. During the measurement, ARR was configured as 72000, twice the CCR value. The pre-scalar will be computed and changed every time when comes to a new frequency.

Note that the load switch is a low-side enable circuit, therefore, the idle state of this PWM signal shall be at the logic **High**. AAR and CCR were initially set to the same value, giving a 100% duty circle to make sure no current flows through the LSA.

Task 3 - Temperature Control

As temperature control is a peripheral function, its control logic of it is very basic and has been carefully illustrated in the previous section 2.4.1. Only the process of ADC reading from the temperature sensor will be provided in more detail below.

Analogue conversion: The temperature sensing PCB embedded with the temperature sensor *TMP236* was designed and populated. More details can be found as an electronics output in the section(Insert link). After reading sensing data from ADC, the digital reading should be converted back to the analog reading for further comparison with the reference temperature (35 °C). The data sheet of *TMP236* gives an equation to calculate the temperature from the output voltage under 100 °C, which is shown in the equation.

$$\text{Temperature} = \frac{V_{out} - 0.5}{0.01} - T_{Calib} \quad (2.5)$$

A calibration test was done before integrating the temperature-sensing PCB into the device. Data shown in the table indicates that there is a 16 °Ctemperature difference between the real temperature and the analog reading from the *TMP236*. Overheating on PCB from the manufacturing process might be the most likely reason for this temperature difference. Therefore, the equation shall be corrected by -16°C (T_{Calib})in the end.

Scenario	Voltage reading/V	Converted Reading/°C	Real Temperature/°C
Room	0.884	38.4	22
Human mouth	1.024	52.4	36

Table 2.1: Calibration data of the *TMP236* sensor

Task 4 - Data storage

UART configuration: The UART on MCU only transmits data for this application. There are three methods to transfer serial data through UART, transmitting data with a single poll, transmitting data with an interruption, and transmitting data with DMA(*Direct Memory Access*). In this case, the

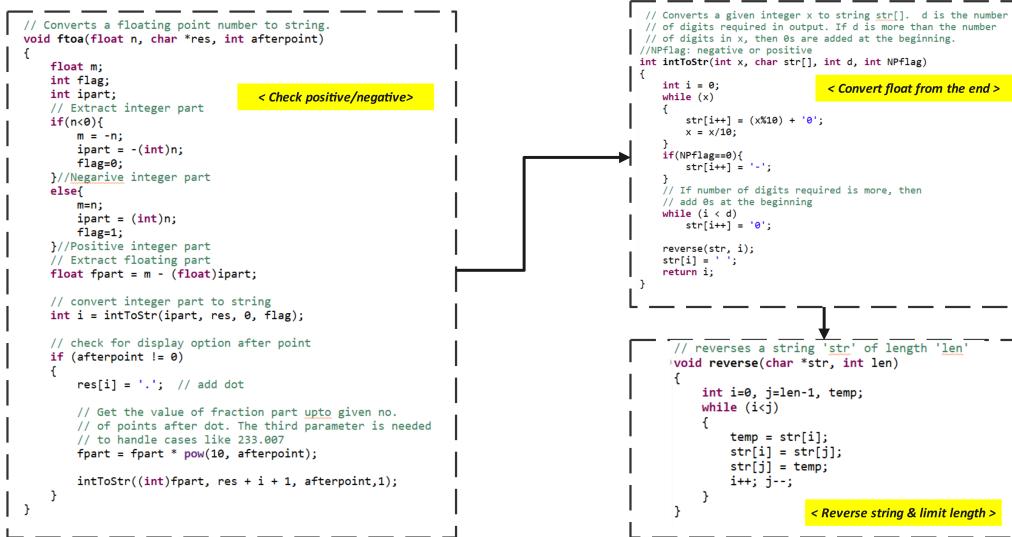


Figure 2.24: Self-written string conversion function

computed resistance and reactance were designed to be transferred through *HAL_UART_Transmit_IT*, the interruption mode of UART, out for two reasons. On the one hand, instead, the CPU blocks other operations every time during UART poll mode, the interruption model allows the transmission to happen without blocking other tasks. On the other hand, UART under the DMA model transfers data without getting through the CPU. However, three DMAs are used with the ADC on other tasks in the meantime, and the size of the measured impedance is fairly small. Therefore, UART under the interruption model was considered more necessary and efficient.

String conversion: Before sending measured data from MCU to the BLE module through UART, a string conversion needs to be applied to convert a floating impedance value to a string. It is because the BLE terminal reads data in either ASCII or binary type.

There were two methods considered for string conversion, the *sprintf* function from the C library or a self-written function. However, current MCU is very code memory size constrained. *printf/sprintf* contributes a good bit to the size of the executable code. It was found that the handler of memory management error was called as long as *sprintf* function was applied. Therefore, a short function was written by the project student herself for an efficient string conversion purpose. Codes with detailed comments were summarised as shown in figure 2.24.

Note that only specific codes are added for the previous illustration because of space limitations. The complete project and programming code can be found on the project student's GitHub linked in *Reference [7]*.

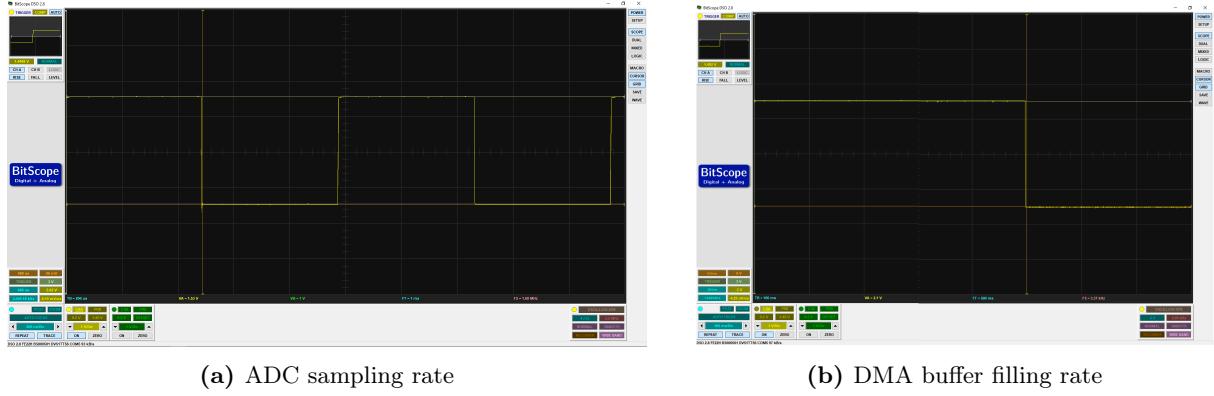


Figure 2.25: Sampling rate of ADC and DMA

2.6.2 Verification

All functions designed in MCU have been verified during the programming process. Apart from the BLE output which has been explained in the previous section 2.5.2, verification results for ADC sampling rate, frequency sweeping output, and FFT function are all carefully illustrated below.

ADC verification

By adding a GPIO output on the ADC complete callback, the sampling rate was measured as shown in figure 2.25a. Oscilloscope indicates the sampling rate of ADC is 2.049KHz, matching the code configuration. DMA reading rate was also measured every time when DMA buffer was full as shown in figure 2.25b. Results show that the DMA buffer is filled up every 0.97s. These results are consistent with the previous discussion for the design stage 2.6.1.

FFT verification

FFT was used on *Task: PQReader* for impedance calculation. The verifying process is provided below:

- Use ADC (Sampling frequency set to 2.048Khz) input a sine wave with frequency 5Hz and offset 1.68V.
- Convert ADC output (Digital value) to Analog value.
- Use *arm_rfft_fast_f32* function (FFT length=2048),get the peak value.

It is known that the FFT for a real N point sequence is even symmetry in frequency response. According to documents provided by *CMSIS DSP library* [11], the output of *arm_rfft_fast_f32* function is shown in the equation 2.6.

$$X = \text{real}[0], \text{imag}[0], \text{real}[1], \text{imag}[1], \text{real}[2], \text{imag}[2] \dots \text{real}[(N/2) - 1], \text{imag}[(N/2) - 1] \quad (2.6)$$

As the result shown in the figure 2.26, the peak frequency response of this sine signal happened on the DC offset (*output[0]*) and frequency response at 5Hz (*output[10]&output[11]*), which proves the functionality of FFT on this MCU.

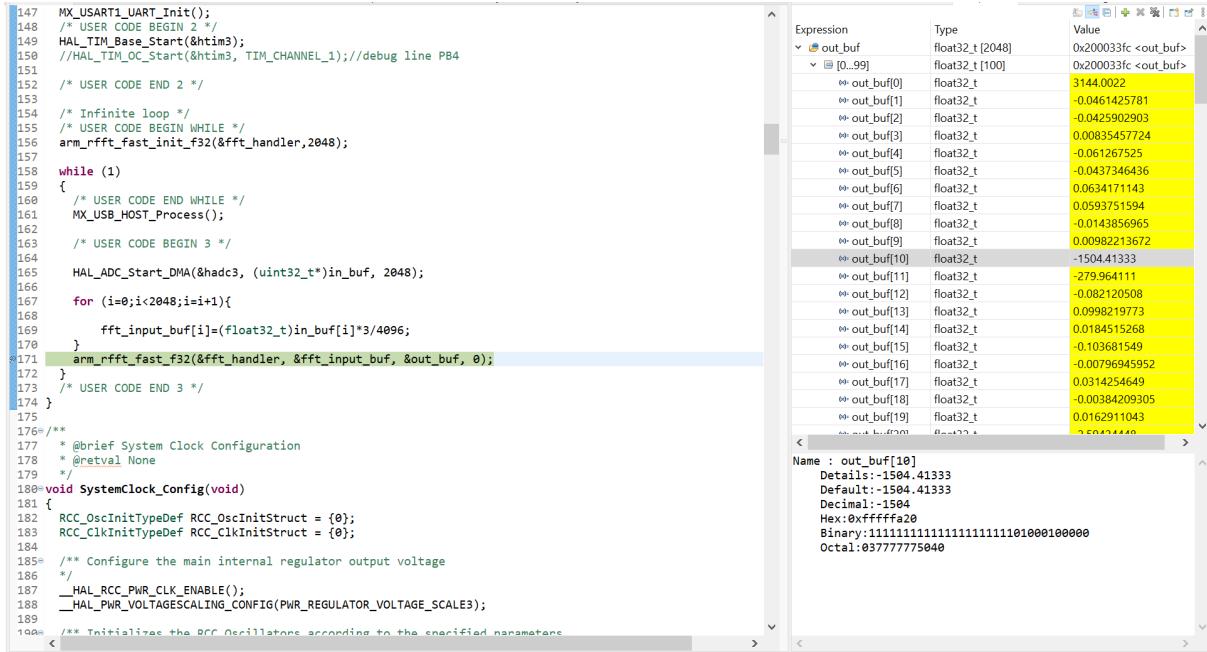


Figure 2.26: Verification result of the FFT on STM32s

Frequency sweeping verification

In fact, the real actuating signal was illustrated in the previous section 2.3.3, which has been good enough to prove both the functionality of the enable signal sent by MCU and the reliability of the load switch circuit. The direct output from *Task: Actuator*, showing how frequency sweeps during the measurement, is given in figure 2.27. Note that this graph is an overlapping graph of separate measurements at different frequencies.

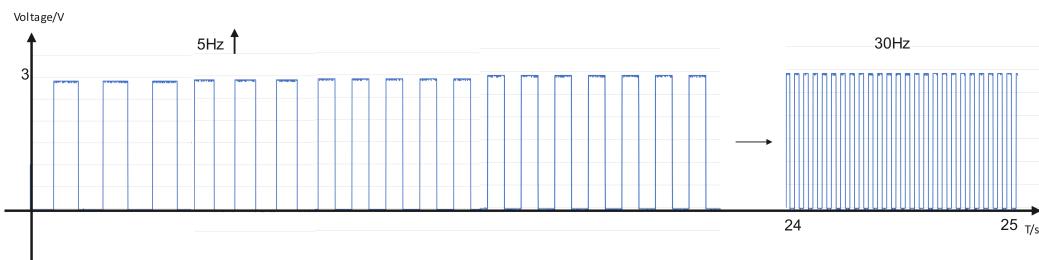


Figure 2.27: Enable signal of the actuating system sent by MCU

Chapter 3

Output and Result

3.1 Output

3.1.1 Electronic Output

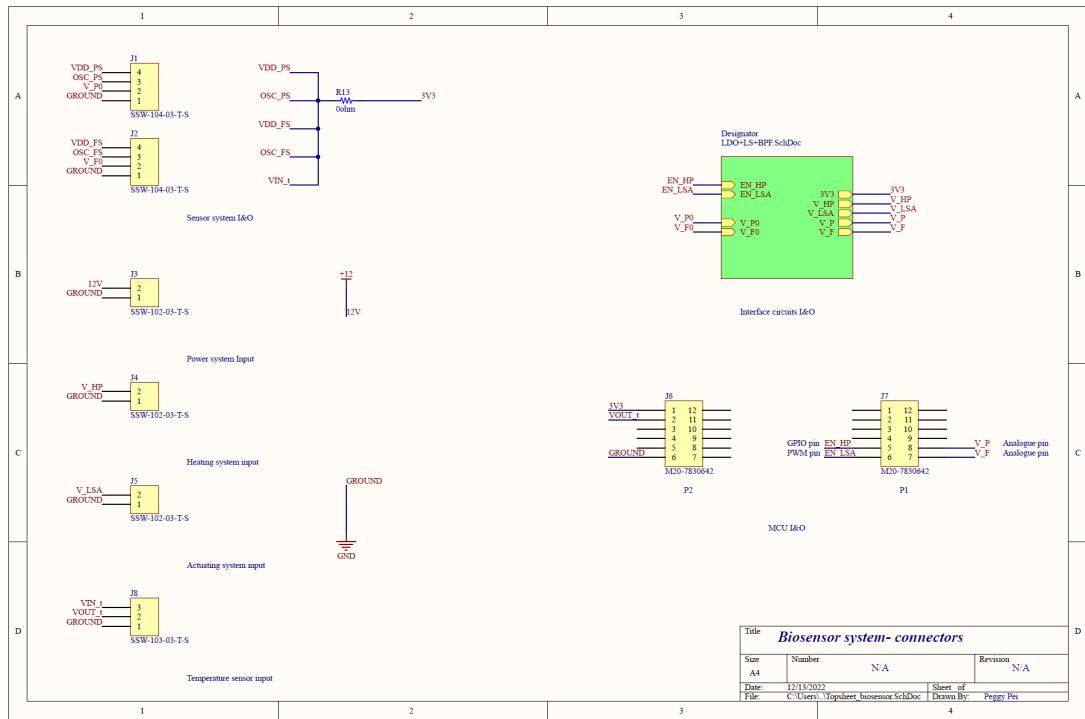
There are two PCBs designed and manufactured through a software called *Altium* for this biosensor system in phase two, the main board and the temperature sensing board. The functionality and PCB layout will be explicitly illustrated for both of them. During the manufacturing process, different versions were updated due to ill consideration from the project student. Each version corrected some mistakes and made some improvements. Details related to each version and any modifications were also summarised below, aiming for providing some reflections for other students on the PCB manufacturing process.

Main PCB

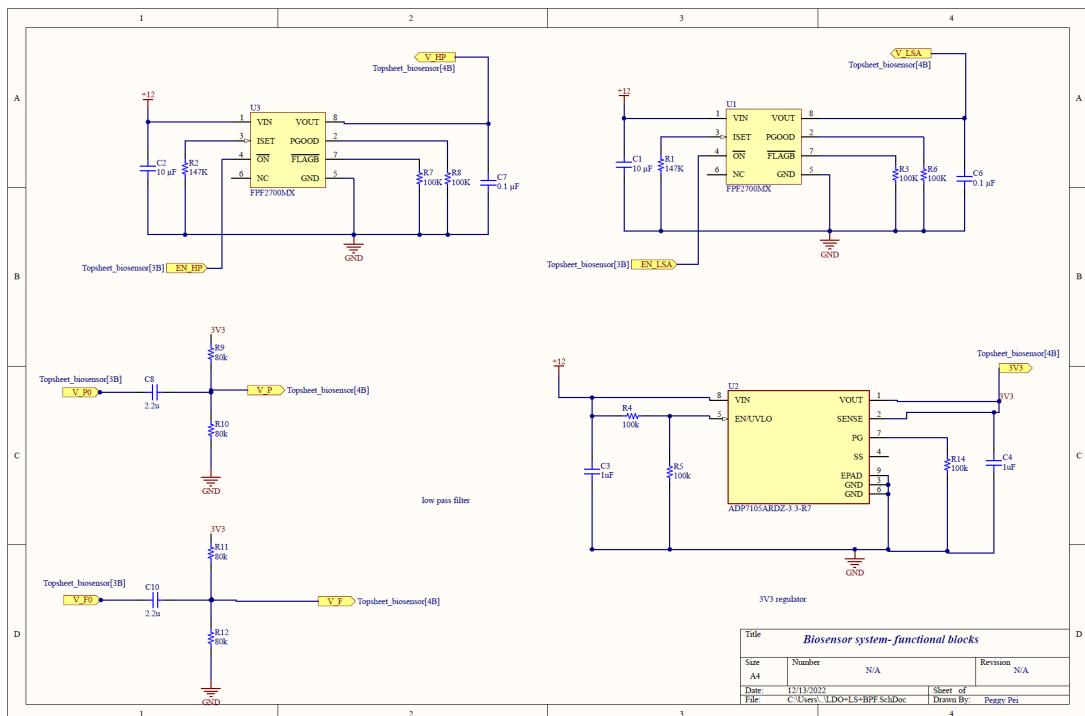
Schematics: The main PCB was integrated with different functional blocks. It is equipped with two high pass filters with a level shift circuit, two load switches for LSA and the heating pad, one 3.3V regulator, and connectors required for all subsystems. Therefore, all blocks are connected through this main PCB to avoid chaotic connections. The schematic of the main board is shown in figure 3.1.

3D PCB layout: The PCB layout for the main board was designed on *Altium*, which is shown in figure 3.2. Note that the size of the main board was designed according to the size of the MCU, which allows the main board to act as a header board of the MCU so that they can be grouped together more neatly.

Version summary: The main PCB has three versions in total. Each version was amended and improved based on the previous version. Features of each version, mistakes including manufacture limitations and ill-considerate designs, and modification related are all summarised in the table 3.1.



(a) Main board - Schematic one



(b) Main board - Schematic two

Figure 3.1: Schematics of the main PCB

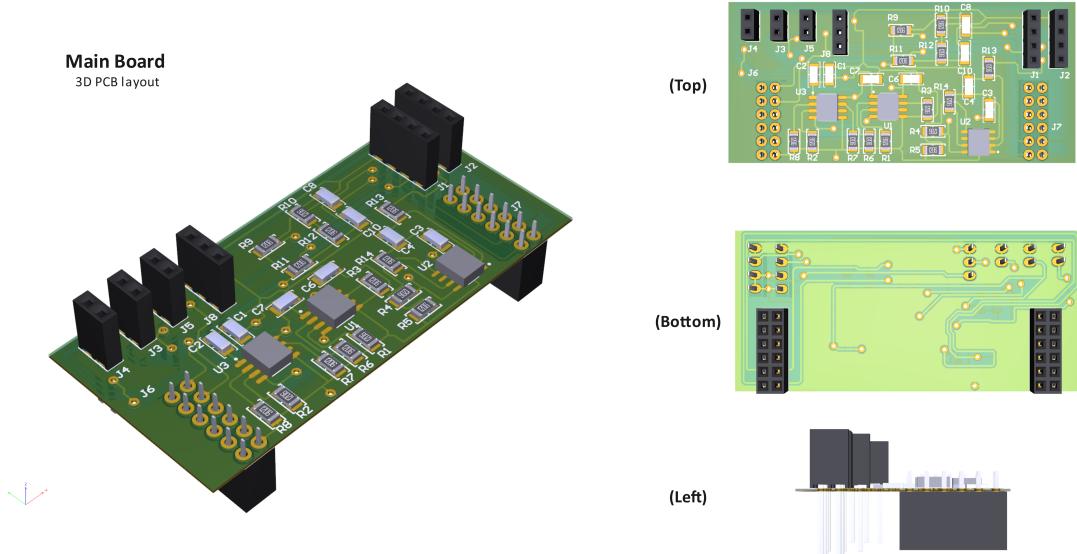


Figure 3.2: 3D PCB layout for the main board

Version No.	Feature	Modification
1	<ul style="list-style-type: none"> -12 Power input -3V3 regulator as power input for the sensor system and MCU -Load switch circuit -Connectors and connection circuits between each subsystem -Ground pour and 12V pour layer. -Bandpass filter for sensing signal 	None
2	<ul style="list-style-type: none"> -Same features as previous 	<ol style="list-style-type: none"> 1. Changed new load switch IC model on Altium(FPF2701MX instead of FPF2701MPX, as they are on the different package.) 2. Delete 12V pour layer (Only a bare board was manufactured. For safety reason and less contact issues, the 12V pour layer was deleted, only the ground pour layer was reserved). However, <u>Ground pour was occasionally deleted by manufacturing technician.</u>
3	<ul style="list-style-type: none"> -Same features as previous. -Change the Bandpass filter to high pass filter with a level shift circuit 	<ol style="list-style-type: none"> 1. Changed 3V3 regulator and its peripheral circuit. (The current limit for previous regulator is 200mA, however, 250 mA is sinking from MCU and 10mA is sinking from sensors. Therefore, change a new 3V3 regulator with current limit up to 500mA) 2. Add vias for through whole connector. (Due to manufacture limitations at the Uni lab, all vias and through whole components have to be manually soldered on both sides, which means through whole connectors were taken apart firstly and assemble back during the soldering process, causing noises and instability. Add vias can avoid this as tracks would be printed at manufacture stage.) 3. Add ground pour layer back. (Solve the manufacture issue mentioned before) 4. Change band pass filter to high pass filter +Level shift (Avoid negative value input to the ADC)

Table 3.1: Version summary of the main PCB

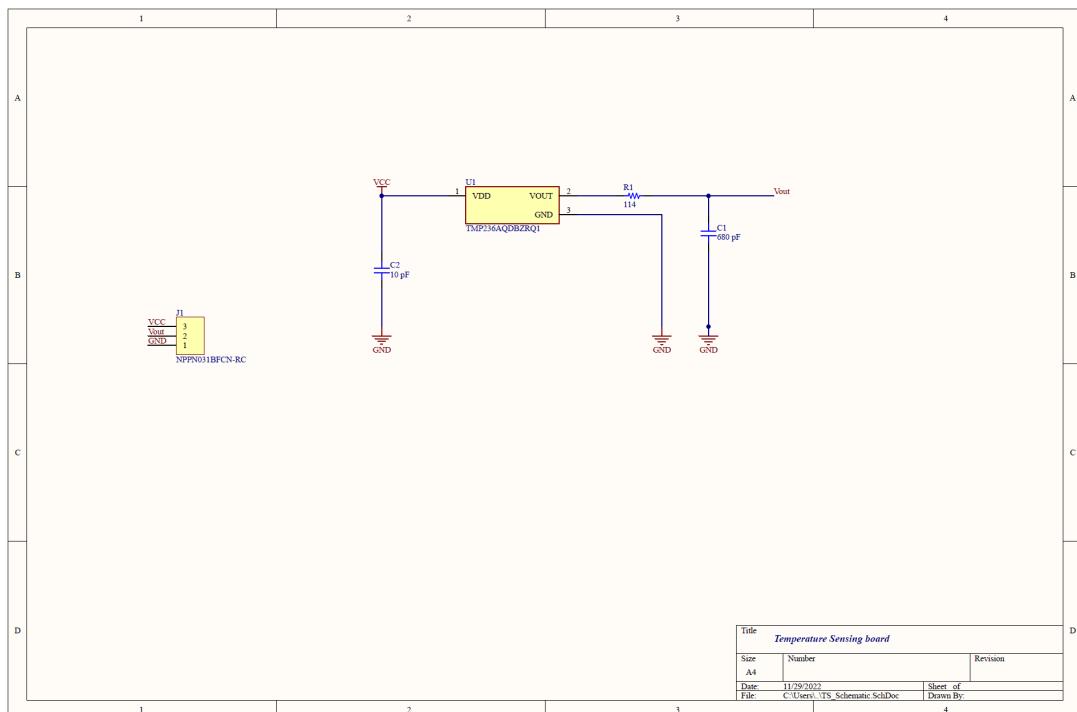


Figure 3.3: Schematics of the temperature sensing board

Temperature sensing PCB

Schematics: The use of temperature sensing PCB has been carefully illustrated as shown in the previous section 2.4.2. The schematic of the temperature sensing board is shown in figure 3.3.

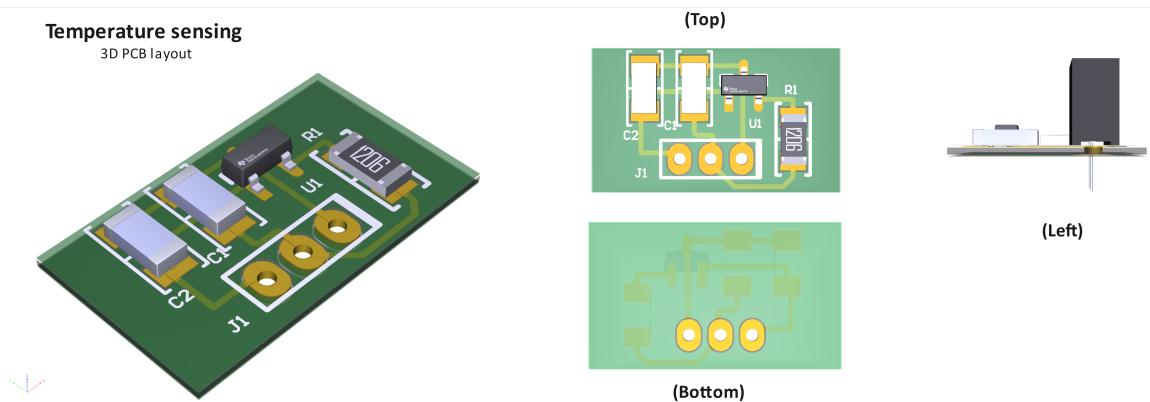


Figure 3.4: The 3D PCB layout of temperature sensing board

Version No.	Feature	Modification
1	-Temperature sensor field. -Low pass filter for temperature sensor output. -Connectors for power input and analogue output.	None
2	Same features as previous	1. Correct layer orientation . (Layer was put in an opposite way by the manufacturing technician) 2. Correct connector length .(A wrong female connector model was used during Altium design stage, the diameter of the connector was manually changed from 2mm to 2.54mm in order to fit the available connector at the lab.)

Table 3.2: Version summary of the temperature sensing PCB

3D PCB layout: The 3D PCB layout of the temperature sensing board is shown in figure 3.4. Note that the size of the temperature sensing PCB is restricted by the length of the connector between the pneumotachometer and mouthpiece. The board was designed to be as small as possible so that it could fit inside the device.

Version summary There are two versions of the temperature-sensing PCB. Due to the manufacturing issue, tracks on board were printed in an opposite way the first time. The connector was also chosen with the wrong diameter. After fixing all problems, the board was reprinted as the second version. Details related are summarised in the table 3.2.

Reflection

In fact, multiple versions were made because of poor consideration and unforeseen manufacturing issues. Mistakes were corrected with the help of technicians at the lab and redesigned carefully by the project student. Therefore, three tips are summarised below as the reflection of PCB application at the beginner level. Hopefully, these will not only be a valuable technical experience for the project student herself, but also a useful reference for the other newbies.

- **Component selection:** The proper component should be considered, evaluated, and simulated at the design stage. One of the problems that occurred was that the current limit of the old 3.3 V regulator was too small. It shall be avoided at the very beginning. Breadboard tests and SPICE simulations are good ways to examine the reliability of the designed circuit before making a PCB and putting any real component on board. What's more, pay attention to the model picked in *Altium Library*. Do be careful when choosing the component as it shall be with not only the right part number but also a compatible package.
- **Copper Pour:** The copper pour is an area on board that is filled with copper as the connection. As shown in figure 3.2, the bottom layer has a ground copper pour. Generally speaking, it is used to make the PCB more robust, more specifically, the prototype with the copper pour always passes the EMC(*electromagnetic compatibility*) test more easily. But when comes to a bare board design, copper pours sometimes are unnecessary and make the manufacturing process more challenging. Do consider carefully before using copper pours to avoid failures of the board.

- **Manufacture limitation:** Without the third party, all PCBs were manufactured by technicians at the lab in the university. It provides fault tolerance for project students. However, there are also many manufacturing limitations. For example, there is no wrapper layer, all tracks are exposed to the air on the manufactured PCB, which makes copper pour a bad choice if the relatively high voltage and current are implemented. Besides, all *Vias*, as the connection between different layers, need to be manually soldered on both sides. It is essential that students shall check the manufacturing standard with the technician first before populating all components on the PCB and testing its functionality.

3.1.2 Mechanical Output

3D CAD: A software called *Onshape* was used to develop the CAD design of mechanisms for the device. The mechanical design delivered three different parts as shown in figure 3.5. Each of them is introduced as follows:

- **PART A) LSA Studio:** a room for the LSA. It has a 2.5 mm height block inside as shown on the top view in figure 3.5, for increasing the stability as the LSA brings a fierce movement during the measurement. It is not sealed on both the top and back sides to achieve easy assembly. Besides, a 70 mm height external support was designed for usability as a measurement device.
- **PART B) Membrane Connector:** a connector between LSA studio and Pneumotachometer. Both membrane and heating pad are set up here. It also has an air outlet for patients to breathe in and out.
- **PART C) PS Connector:** a connector between pneumotachometer and mouthpiece. It is the place to measure the pressure and settle down the temperature sensing PCB as shown on the back view in figure 3.5. Similarly, this connector also has external support for the same reason mentioned before.

The dimension of each mechanical part was calculated based on each fit-in component such as the LSA, the pneumotachometer, the thickness of the membrane and etc. Due to the page restriction, more details are given in the mechanical drawing shown in appendix A.

Version Summary: There are two versions of the mechanical design. Only *PART A) LSA Studio* was designed and manufactured twice. It is because the tolerance for the junction area between the LSA studio and the membrane connector was set to 0.1 mm, which is too small to clamp them together. Note that 0.2 to 0.4 mm is often required as the tolerance for two interlinked components. There are also other small modifications on LSA studio to make the LSA fit in more easily and firmly. Details have been summarised in the table 3.3.

3.1.3 Integration

The real configuration of each subsystem has been clarified before in Chapter two when every function block was separately demonstrated, therefore, only the overall configuration of the device and MCU will

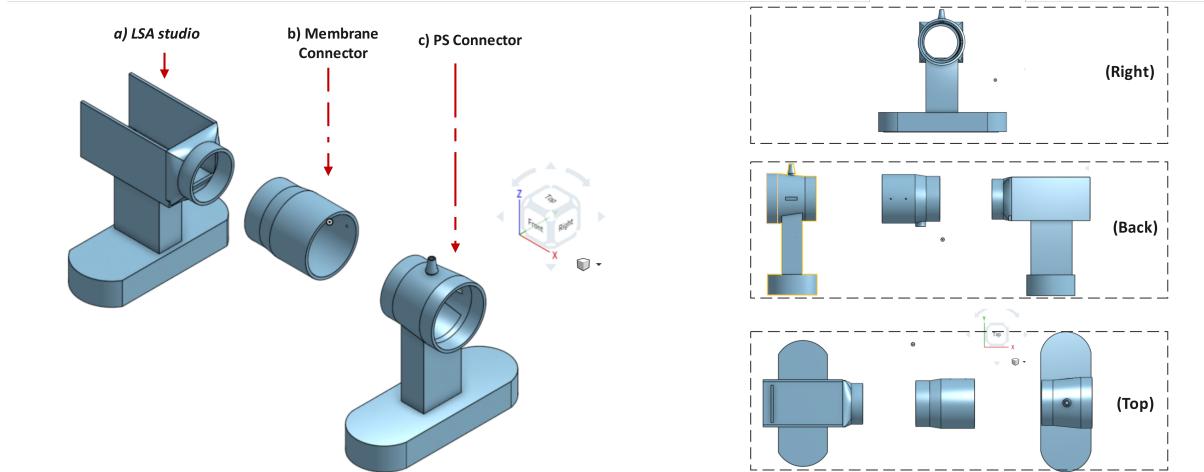


Figure 3.5: The 3D mechanical output

Version No.	Feature	Modification
1	<ul style="list-style-type: none"> -Part 1: LSA studio -Part 2: Membrane connector(Connect Membrane+ Air outlet) -Part 3: Sensor connector (Connect pressure sensor and temperature sensor) 	Sand the edges of the junction.
2	<ul style="list-style-type: none"> -Same features as previous 	<p>Only for Part A LSA studio:</p> <ol style="list-style-type: none"> 1. Opened the top roof, providing more space for cable settlement of LSA. 2. Reduced the stroke length for LSA(12cm to 14.3cm) in order to attain a higher force 3. Changed the diameter of Membrane junction. (Previous tolerance for the junction area between two parts was set to 0.1mm, which brought difficulties for assembly. Note that 0.2-0.4mm tolerance is often required if a tight connection is expected.)

Table 3.3: Version summary of the 3D mechanisms

be established here as the integration output.

Dimension: The overall dimension of this integrated device is in $279\text{mm} \times 90\text{mm} \times 100\text{mm}$ ($\text{Length} \times \text{Width} \times \text{Height}$), measured manually, which should make it acceptable as a portable device.

Configuration: There are two parts of the biosensor device as the output at phase two, the device and MCU ICs. The main device contains the actuating system, the sensing system, the heating system, and a mouthpiece, while the MCU consists of the MCU, the main PCB, and the BLE module. The configuration of those two parts is shown in figure 3.6 and 3.7 respectively. Subsystems are integrated based on the schematics and block diagrams that were carefully explained before. Appendix B and C provides more information like I&O of each subsystem and pinout for PCBs and MCU, which shall be a useful reference for integration of this device. More pictures of the cryptic configuration of the device can be found in the appendix D.

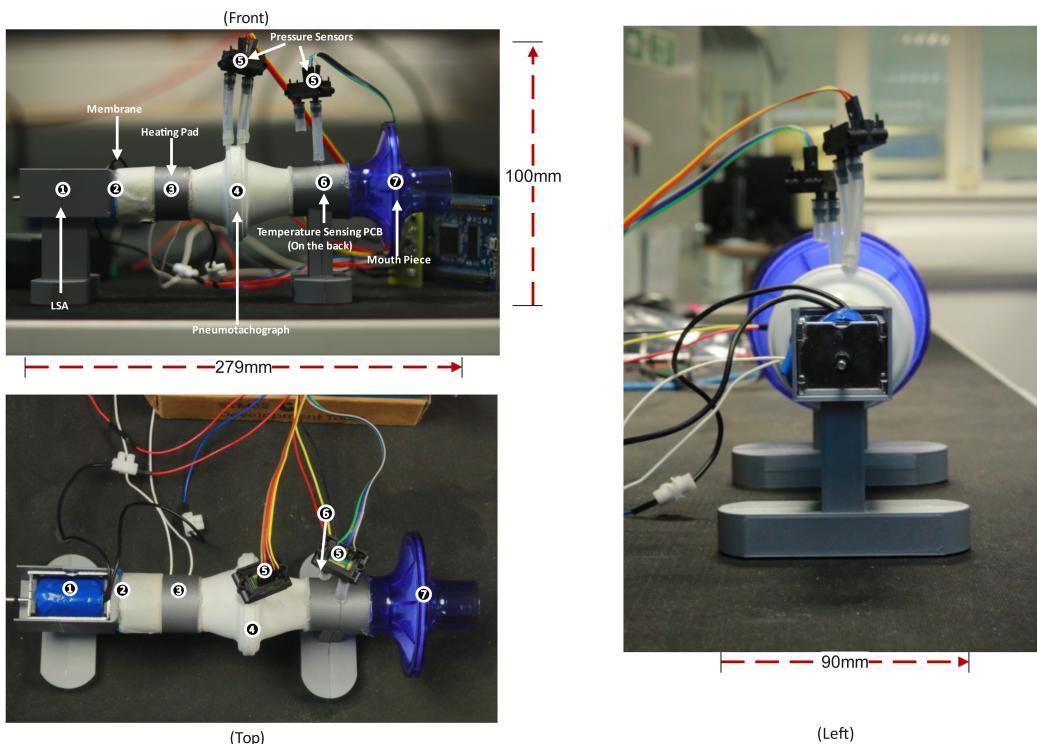


Figure 3.6: Integration-device



Figure 3.7: Integration-MCU

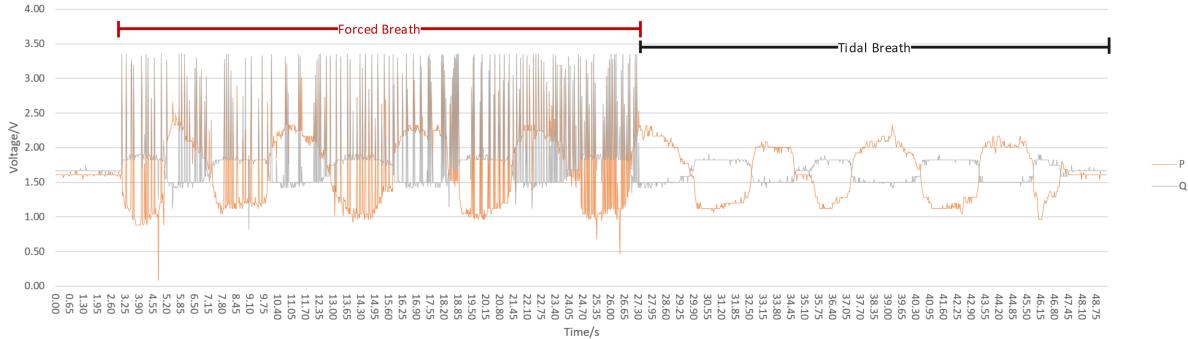


Figure 3.8: Measured breath in a single measurement

3.2 Result

As an experimental subject, both the breath and respiratory impedance were measured by the project student after the final integration of the device. However, there is only one rough measurement was done and analyzed here because of a tight time constraint at phase two. The experimental results at this phase and the simulation results at phase one are presented with the comparative analysis as follows.

Measured breathing: The forced breath and tidal breath were measured in the same process as shown in the figure 3.8. The actuating signal was superimposed on the patient's tidal breath from 2.95 s to 27.95 s while the measurement started. After 27.95 s, the LSA stopped generating signals, sensors captured the tidal breath instead. It is clearly shown that the pressure and flow rate of the patient's tidal breath were in phase and those of the forced breath were out of phase due to the overlapping of actuating signals, which is consistent with previous research.

Measured Impedance: The impedance was calculated by the MCU and collected from the workspace in *STM32CubeIDE*. The real-time impedance-frequency plot is shown in figure 3.9a. It offers that the measured respiratory resistance floats around a positive value as expected. However, the initial resistance at 5Hz is negative. It might be the reason that it takes time for the system to become dynamically stable. Therefore, the initial point of the sweeping frequency should be smaller than 5Hz as the important information is carried at this frequency, which requires the measurement to be in a steady stage. The ideal RX plot given by simulation of the aRIC model at phase one is shown in figure 3.9b. Compared to the ideal reactance, measured respiratory reactance fluctuates with both negative and positive values in an undesired way. The clear relationship between the respiratory reactance and the sweeping frequency is not known from figure 3.9a. In conclusion, the results indicate that the respiratory impedance measured by this current device is not accurate as it is not compatible with the previous simulation result. There are two reasons considered to be factors impacting the final impedance computation.

- **Improper working range of the pressure sensor.** As shown in the figure 3.8, the pressure and flow rate were both up to 3.3 V which exceeded the maximum working voltage of the sensor

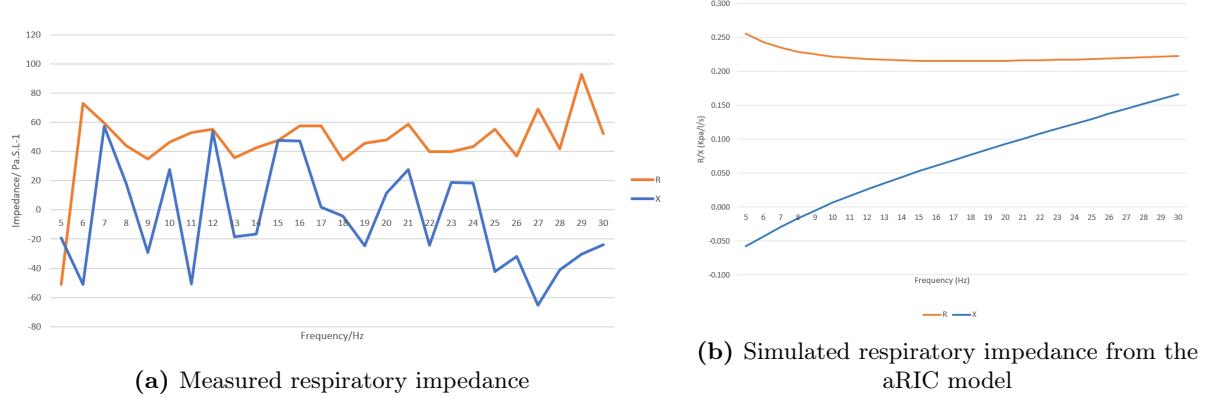


Figure 3.9: The respiratory impedance *VS* frequency

SDP816 when the forced breath was measured. Due to the collision of two air currents of the same large size, the pressure sensor should be equipped with a bigger operational range than it was expected. This is probably a crucial reason that makes the impedance inaccurate.

- **Improper computation logic designed in MCU.** The computation process was designed that the frequency of the actuating signal sweeps from 5 Hz to 30 Hz at 1 Hz per second. It means that MCU allocates and calculates sensor data at this specific second and passes the computed impedance to the memory. However, the tidal breath is often at 0.2 Hz–0.5 Hz, and sweeping time is far shorter than a breathing cycle. From the previous study, the sweeping time is recommended to be set as 1.5 s to 2 s. There is no obvious evidence showing the relationship between the sweeping time and the accuracy of measured impedance. But it is reasonably doubted that insufficient response time may be one of the reasons for inaccurate measurement results. Alternatively, the impedance can be computed when the sensing data are allocated after the actuating signal has swept all frequencies. However, it brings a big issue of memory management for a small microprocessor to store the amount of sensing data over a long period of time. Following modification methods on the computation logic are worth to be researched at a further stage.

Chapter 4

Future Work

Due to the time constraint of the second phase, only the design and verification work of the current system was completed. However, there is still plenty of future work that needs to be processed if the biosensor device is expected to achieve a more reliable performance as a medical device on the market. Instead of considering the future development required to commercialise a medical device, like regulatory inspections or marketing strategies, this chapter will continuously focus on the R&D process of this biosensor device, providing more potential approaches to improve the functionality of the device based on subsystem units.

4.1 Power System



Figure 4.1: EVAL-SCS001V1 board

USB power delivery protocol Initially, the USB power delivery protocol was carefully considered. There are many advantages to it. By choosing the right IC with the USB PD feature, different power states can be achieved at once. For example, EVAL-SCS001V1 (Fig. 4.1) embedded with IC **STUSB4500** is from the ST-Microelectronics family. It is a development board pre-configured with 3 power profiles, 5 V/1.5 A, 15 V/1.5 A, and 20 V/1 A. It allows users to do a fast migration to USB type C connectors for their applications. Alternatively, up to 100 W (20 V/5 A) can be achieved through its USB PD sink port. The board can be customized by coding at a further stage.

The reason why the USB PD protocol was not used on this project at this phase is that it indeed depends on the high coding level of the developer. Power IC has to be properly communicated with the MCU,

which requires them to be under the same electronics family, programmed by the same language but may be on a different compiling platform. Besides, the USB PD sink port needs to be connected to another USB source device like a power bank, a PC, or a laptop, which may be restricted by their own power delivery rate. The high coding technique is essential for developers to be able to eventually match the power delivery rate among different electronic devices. However, it is still an approachable method to manage the power system of this device. Extra conversion circuits like 3.3 V regulators and load switches could be possibly deleted if power states can be varied and controlled by USB PD IC properly.

4.2 Sensing System

- Test the real HPF with a level shift circuit on the PCB.
- Consider active filter design to achieve a more actuate filtering performance.

4.3 Actuating System

- Consider methods to reduce the harmonics effect of the actuating signal. For example, designing a proper low pass filter circuit to remove harmonics at undesired frequency.
- Consider methods to reduce the amplitude of actuating signals and make sure forced breath can be sensed within the operational range of sensors. Thickening the membrane and increasing the tension of the membrane could be practicable approaches.
- If stay with the current actuating system, recheck and recalculate the sensing range by referring to the collected data of forced breath. To make results more readable, change a new sensor with a bigger resolution if it is necessary.

4.4 Heating System

- Design a new control logic using two sensors as the feedback, avoiding issues like overheating and sensing delay as discussed in the previous subsection 2.4.3.
- Develop the PID controller to compensate for temperature more precisely and intelligently.

4.5 Communication System

- Design a keyboard on touchscreen GUI for users to input their personal information.
- Modify the *Save Data* function on MCU to solve the current problem of wireless communication.
- Alternatively, consider other approaches to transfer the measured data. For example, a USB-C cable can be connected between MCU and a tablet. *STM32F429I-DISC1* discovery board is equipped

with the **USB-OTG-FS** protocol and has a user USB-C port for wired communication. But it was unavailable when a GUI was implemented at the same time. It was because GUI uses the FMC (*Flash Memory controller*) to manage large amounts of memory. Specific pins of the FMC function were restricted on board. However, theoretically, the default settings of the pin can be modified through high-level programming.

4.6 Central Control System

- Synchronization issues between different tasks are currently using the single queue in FreeRTOS. However, there are more advanced message mechanisms like locking mechanism *mutexes* and signaling mechanism *semaphores*, which can make messages transferred in a safer and better-organized way.
- There was aRIC model (*The augmented RIC model*) built to serve as supplementary quantitative approaches for diagnosing respiratory diseases at Phase One. Simulation results gave a reliable prediction of the computed respiratory impedance. Because of the time restriction, the final computed impedance was not able to be measured and analysed with this current device. But It would be massively useful if the aRIC simulation model can be fed into the control system to first verify the reliability of the current MCU design as a next step.

Chapter 5

Impact and Exploitation

5.1 Research and Development Process

The research and development process of the biosensor device is designed and shown in figure 5.1. It derives from the Gantt chart set in phase one [1] and contains more information about further development steps required to commercialise a medical device into the market. The whole process is divided into three different stages, the early consideration stage, the product development stage, and the product launch stage.

Stage I - Early consideration In the early consideration phase, an engineering idea aiming to develop a portable device for the daily respiratory monitoring of COPD patients kicked off this project. The targeted users, the purpose of this product, and the budget were defined accordingly. To further process the concept into reality, an amount of research related to the technique, market as well as revenue was conducted before jumping off the design stage. It is worth noting that the QMS(*Quality Management System*) shall be properly set up at the beginning for the development of any medical device as asterisk commented in figure 5.1. As what typically distinguishes medical devices from other products is that they are subject to more strict regulations depending on different categories, classifications, and launched countries. However, the regulatory framework of this biosensor device was not built due to the Complexity and sub-relevance of regulations. The project focus only on the R& D process on the engineering side. But it is still an essential step to commercialise a medical device. *ISO 13485* standard, addressing the development, implementation, and maintenance of a quality management system intended for use by medical device manufacturers and suppliers, is commonly used for regulatory purposes in the UK market [12].

Stage II - Product development The leading design work was delivered at this stage, including both system design and simulation in phase one and practical output from the design transfer in phase two. The red flag shown in figure 5.1 represents the current state in the R&D process of the biosensor device. A prototype has been successfully produced at the end of phase two. 80 % functionality has

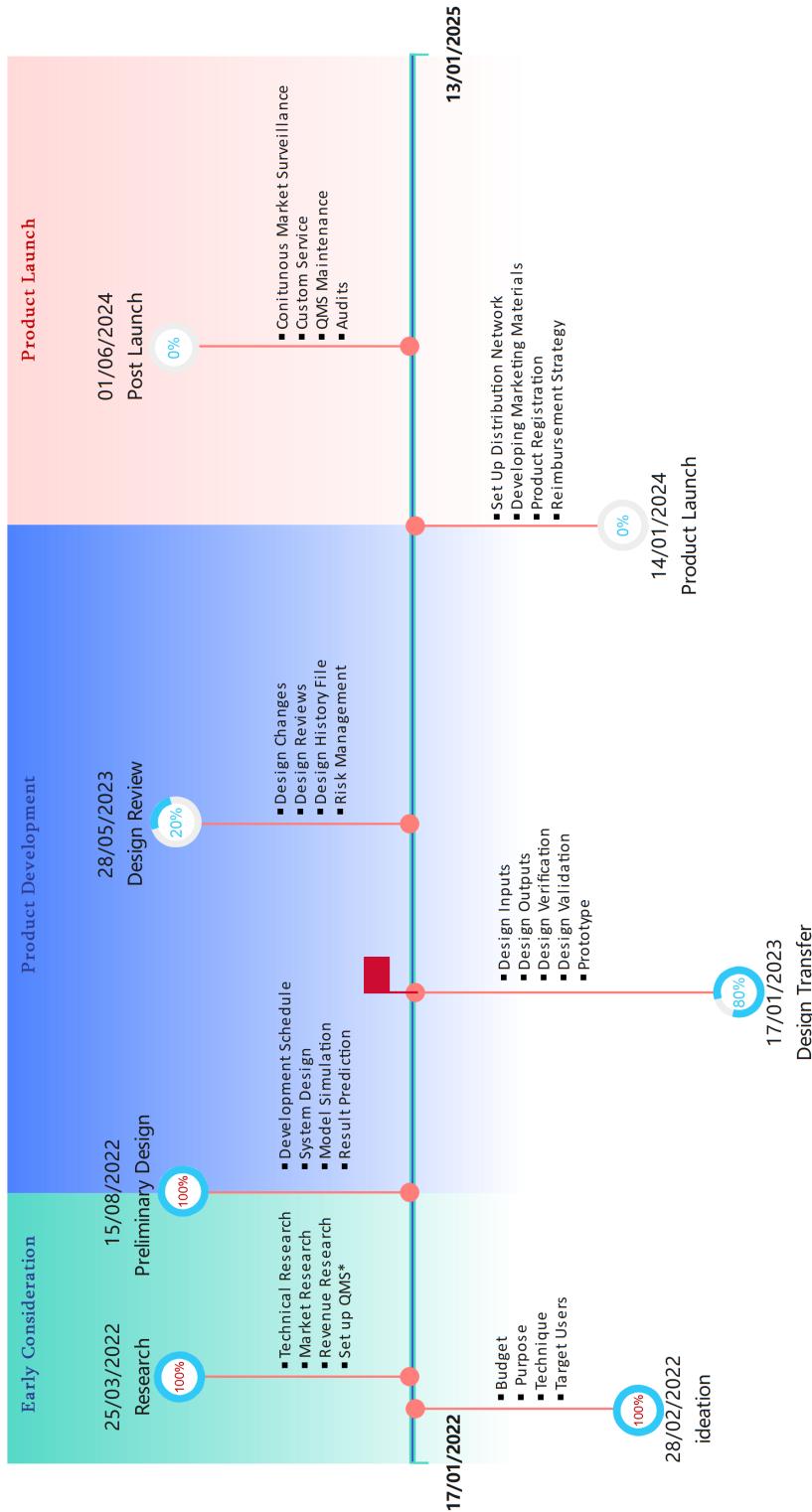


Figure 5.1: Research and Design Process of the biosensor device

been achieved after preliminary design. However, there are still design reviews required for potential improvements and modifications of the device, in order to let it conduct a reliable measurement.

Stage III - Product Launch Here is the final stage for the biosensor device. Preparation work like developing market materials, registering the product, and establishing marketing strategies, is required before the product launch. When the product is floated on the market, tight supervision of the device is necessary regardless. This is the basis to let customers rest assured to choose and make the product long-lasting and evolvable.

5.2 Impact

Social impact From the 2023 Gold Report published by the *Global Initiative for Chronic Obstructive Lung Disease* [13], the official estimation for the global prevalence of COPD is 10.3 % (8.2 %,12.8 %) in 2022. COPD has led to high morbidity and mortality worldwide with a growing economic and social burden. Therefore, there's a sense of urgency for the COPD management from a social perspective. Follow-up supervisions and daily monitoring, especially informative measurements obtained from pulmonary function tests, are essential for COPD management according to the 2023 GOLD report [13]. This biosensor device offers an innovative way for the COPD management in the future, providing positive social impact in many aspects.

On the one hand, the portable device improves the accuracy of professional diagnosis and treatment from physicians. The GOLD guideline addresses the pulmonary function measurement as the gold standard for COPD diagnosis [13]. But a study researching on the COPD in China concluded that less than one-third COPD diagnoses were made with the aid of regular pulmonary function measurement like spirometry. The lack of precise measurement resulted in poor physician recommendations, especially for pharmacologic therapies for COPD [14]. However, this portable device provides both simple maneuverability and high-accurate measurement, which shall be able to play an important role in COPD management and supervision.

On the other hand, this portable device is conducive to raising public awareness of COPD, aiming to orient a healthier lifestyle for COPD patients. The same study as mentioned before emphasised that most COPD patients in China did not visit a doctor until they developed significant symptoms and exacerbations. Besides, only 6.5 % of patients had ever received the regular lung function test [14]. This is because the spirometry device is often a piece of giant and professional medical equipment settled down in the hospital and can be barely approachable in daily life. Patients have limited resources to attain further understanding of respiratory measurement as well as pulmonary diseases. However, this portable biosensor device, using FOT, is extremely easy to carry anywhere at any time. It is unlikely the normal spirometry device that requires lots of effort from patients during the measurement [1], which also makes the regular monitoring for patients easy to cope with. Therefore, the portable device is expected to be an efficient tool not only for the daily monitoring of COPD patients but also to raise more awareness for people to know the importance of COPD management and prevention.

Economics impact According to a UK-based study, the annual average cost in only primary care for a COPD patient is up to £2108 [15]. However, previous research highlights the cost of second care for COPD patients, mainly about follow-up hospital care, normally contributes even more compared to that of primary care, accounting for 54% of the direct annual cost [16]. Those facts indicate that COPD places a significant burden on both the health system and COPD patients themselves.

This portable device was developed, aiming to provide a low-cost solution for COPD patients to do regular monitoring. It not only has the potential to give a wide profit margin on the business side, compared to other marketed FOT devices that are extremely expensive such as **Resmon PRO FULL (V3)** [17] and **PULMOSCAN** [18]. What's more, the competitive price allows the patient to be able to do the daily measurement, keeping a good eye on their health condition. It is reasonably predicted that the frequency of hospital care or visit will decrease significantly after patients start using this device, massively alleviating the economic burden of COPD.

Environmental impact Portability is one of the most remarkable features of the device. On the one hand, smaller volume device reduces waste during the manufacturing process. With sustainable materials selected, the portable device intends to be an environmental-friendly product. On the other hand, the portable device makes COPD monitoring possibly happen without the limitation of time and place. Most of the time, patients do not have to travel to the hospital, which also reduces the environmental impact of traffic.

Acknowledgements

This one-year MEng project started in Jan 2022 and ends here. In the past year, I received a lot of help and guidance from different people in different fields.

I'd like to particularly show my gratitude to my supervisor Dr. Stewart Smith, who insisted on meeting with me every week, provided me with useful advice, and gave me space to question, think diversely as well as make mistakes.

Also, my intern experience at *OSSUR-Touch Bionics* company last summer played a vital role in this project. I learned massively about the difference between a project held by the university and business. My manager Phil and colleague Mark provided me with not only technical inspiration but also mental encouragement, which I appreciate all the time.

Lastly, I'd like to thank electronics lab technicians Iain Gold, Alasdair Christie, Alan Robertson, and mechanical lab technician Calum Melrose, who generously provided help whenever I needed it. They bore me as a premature engineer and helped me with manufacturing as many times as I requested. I appreciate their patience and support very much.

This might be the end of this project, but it is never an end for me as an engineer. I would keep learning, questioning, and bearing those kindnesses in mind, to push myself to be a qualified bioelectronics engineer in the future.

References

- [1] Pei, P.: ‘MEng Project Phase 1 Report “A portable bio-sensing device- Measuring the respiratory impedance for COPD patients”’, University of Edinburgh, <https://www.pftforum.com/blog/pneumotach-accuracy/>.
- [2] wiki: ‘Graphical user interface’, [https://en.wikipedia.org/wiki/Graphical_user_interface#:~:text=The%20GUI%20\(%5C%2F%5C%CB%5C%8Cd%5C%CA%5C%92i%5C%CB%5C%90,command%5C%20labels%5C%20or%5C%20text%5C%20navigation.,](https://en.wikipedia.org/wiki/Graphical_user_interface#:~:text=The%20GUI%20(%5C%2F%5C%CB%5C%8Cd%5C%CA%5C%92i%5C%CB%5C%90,command%5C%20labels%5C%20or%5C%20text%5C%20navigation.,), accessed Dec. 2022, Wiki.
- [3] TouchGFX: ‘Model-View-Presenter Design Pattern’, <https://support.touchgfx.com/4.14/docs/development/ui-development/software-architecture/model-view-presenter-design-pattern>, accessed Dec. 2022, website owner TouchGFX.
- [4] Abhimanyu Panditt: ‘How to Use HM-10 BLE Module with Arduino to Control an LED using Android App’, <https://circuitdigest.com/microcontroller-projects/how-to-use-arduino-and-hm-10-ble-module-to-control-led-with-android-app>, accessed Dec. 2022, website owner Circuit Digest.
- [5] DSD TECH: ‘HM-10 instruction’, <http://www.dsdsotech-global.com/search/label/hm%5C%2010>, accessed Dec. 2022, website owner DSD TECH official.
- [6] FreeRTOS: ‘Tasks and Co-routines’, <https://www.freertos.org/taskandcr.html>, accessed Dec. 2022, FreeRTOS.
- [7] Pei, P.: ‘Biosensor device design-Peggy Pei- GitHub files’, <https://github.com/Peggy-p0799/Biosensor-UI-MCU>, accessed Dec. 2022, P.P.
- [8] sensirion: ‘Analog DP sensor (± 500 Pa), tube connection’, <https://sensirion.com/products/catalog/SDP816-500Pa/>, accessed Dec. 2022, sensirion.
- [9] Harvard Apparatus: ‘Respiratory Flow Heads’, <https://www.adinstruments.com/products/respiratory-flow-heads>, accessed Dec. 2022, ADInstruments.
- [10] STMicroelectronics: ‘Application note: STM32 cross-series timer overview’, https://www.st.com/resource/en/application_note/dm00042534-stm32-crossseries-timer-overview-stmicroelectronics.pdf, accessed Dec. 2022, STMicroelectronics.

- [11] CMSIS DSP Software Library: ‘Real FFT’, https://www.keil.com/pack/doc/CMSIS/DSP/html/group__RealFFT.html#details, accessed Dec. 2022, CMSIS.
- [12] ISO: ‘ISO 13485 - Quality management for medical devices’, <https://www.iso.org/iso-13485-medical-devices.html>, accessed Dec. 2022, ISO.
- [13] Chronic Obstructive Lung Disease, G. G. I. for: ‘GOLD-2023-ver-1.1-2Dec2022WMV’, in: (), URL: <https://goldcopd.org/2023-gold-report-2/>.
- [14] Fang, X., Wang, X. and Bai, C.: ‘COPD in China: The Burden and Importance of Proper Management’, in: *Chest* 139 (4 Apr. 2011), pp. 920–929, ISSN: 0012-3692, DOI: 10.1378/CHEST.10-1393.
- [15] Punekar, Y. S., Shukla, A. and Müllerova, H.: ‘COPD management costs according to the frequency of COPD exacerbations in UK primary care’, in: *International Journal of Chronic Obstructive Pulmonary Disease* 9 (Jan. 2014), p. 65, ISSN: 11782005, DOI: 10.2147/COPD.S54417, URL: /pmc/articles/PMC3890403/%20/pmc/articles/PMC3890403/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3890403/.
- [16] Britton, M.: ‘The burden of COPD in the U.K.: results from the confronting COPD survey’, in: *Respiratory Medicine* 97 (SUPPL. C Mar. 2003), S71–S79, ISSN: 0954-6111, DOI: 10.1016/S0954-6111(03)80027-6.
- [17] MGC Diagnostics: ‘The Resmon PRO FULL (V3)’, <https://mgcdiagnostics.com/products/resmon-pro-v3-forced-oscillation-technique>, MGC.
- [18] PulmoScan: ‘PulmoScan oscillometre’, <https://pulmoscan.cognitalabs.com/>, pulmoscan.

Appendix A

Mechanical Drawings and Dimensions

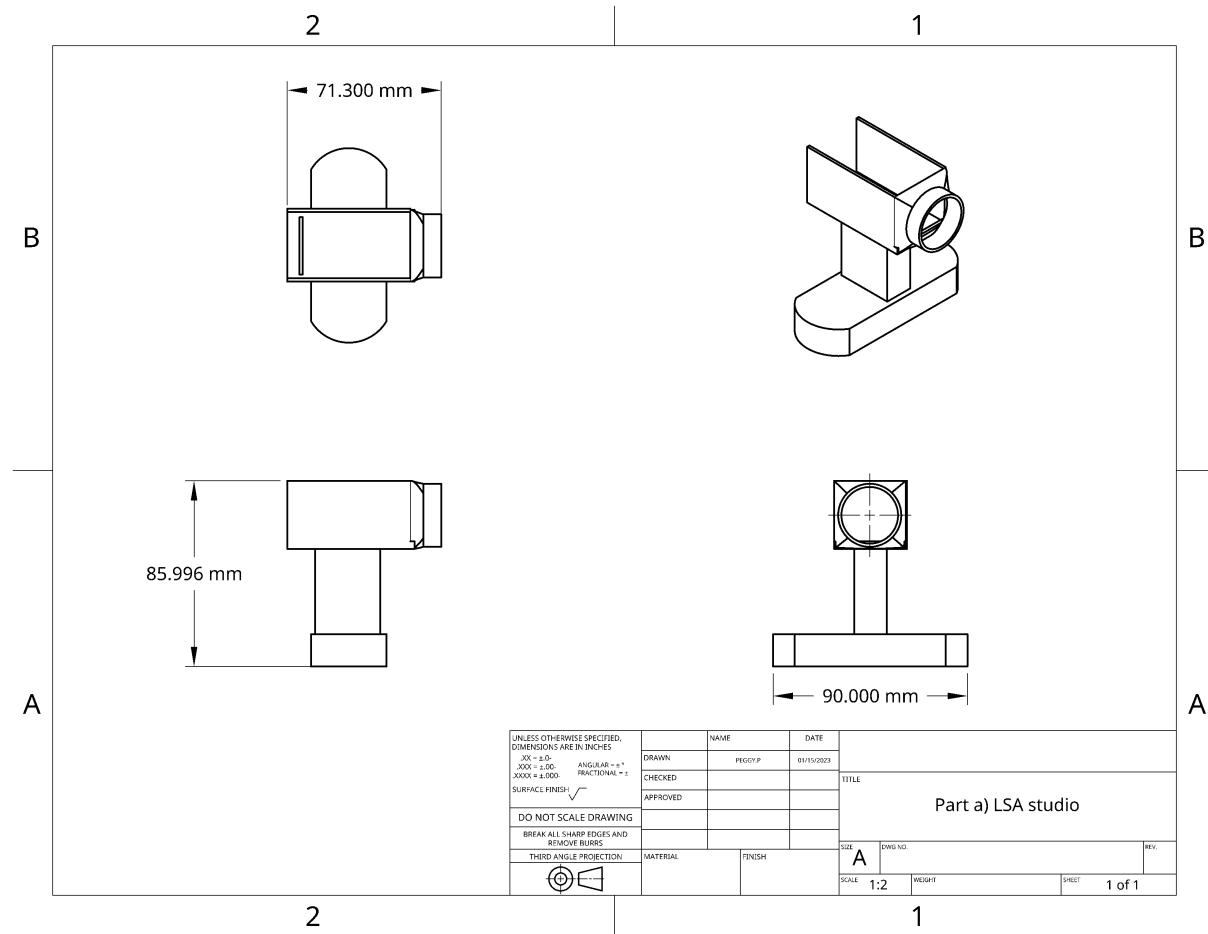


Figure A.1: Mechanical drawing with dimensions - Part A LSA studio

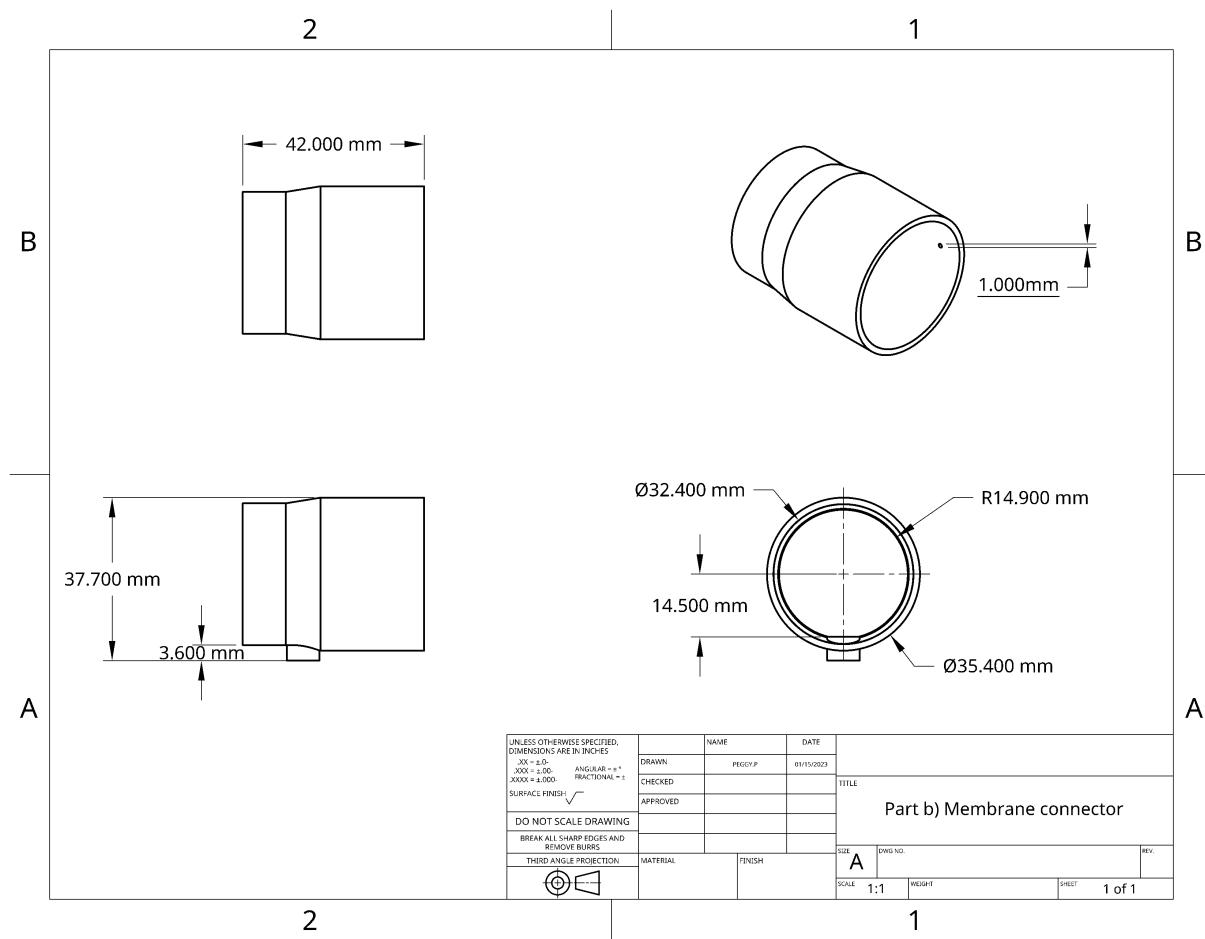


Figure A.2: Mechanical drawing with dimensions - Part B *Membrane Connector*

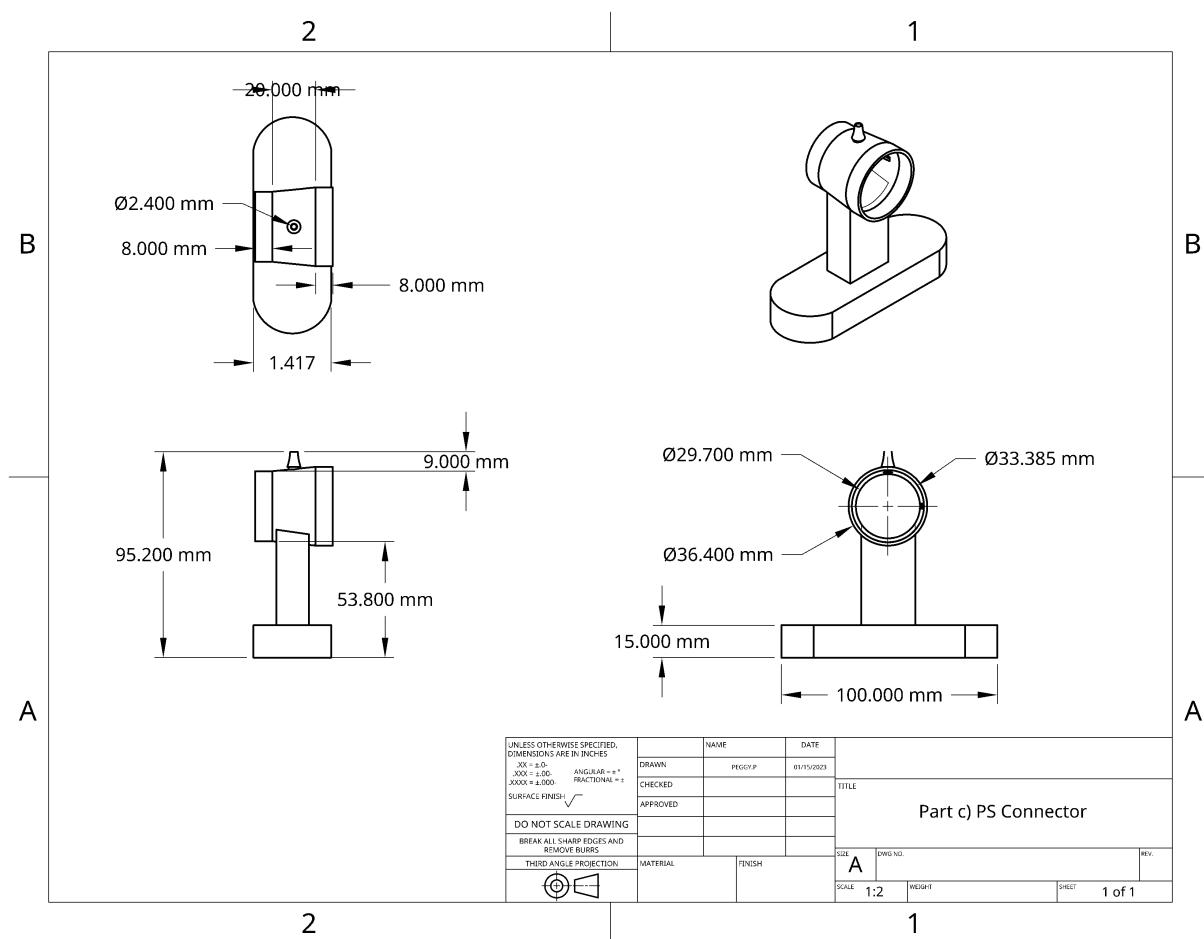
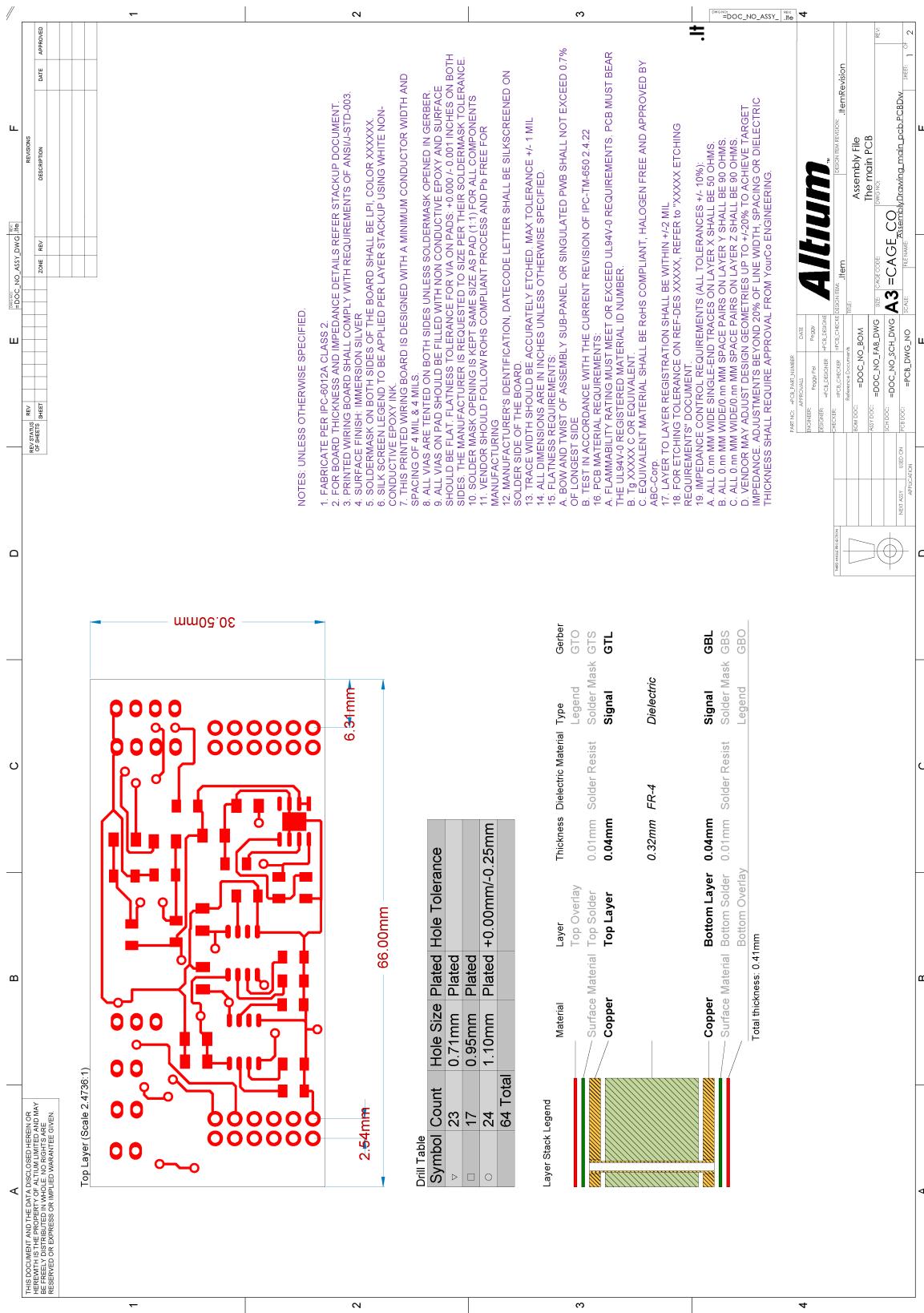


Figure A.3: Mechanical drawing with dimensions-Part C PS Connector

Appendix B

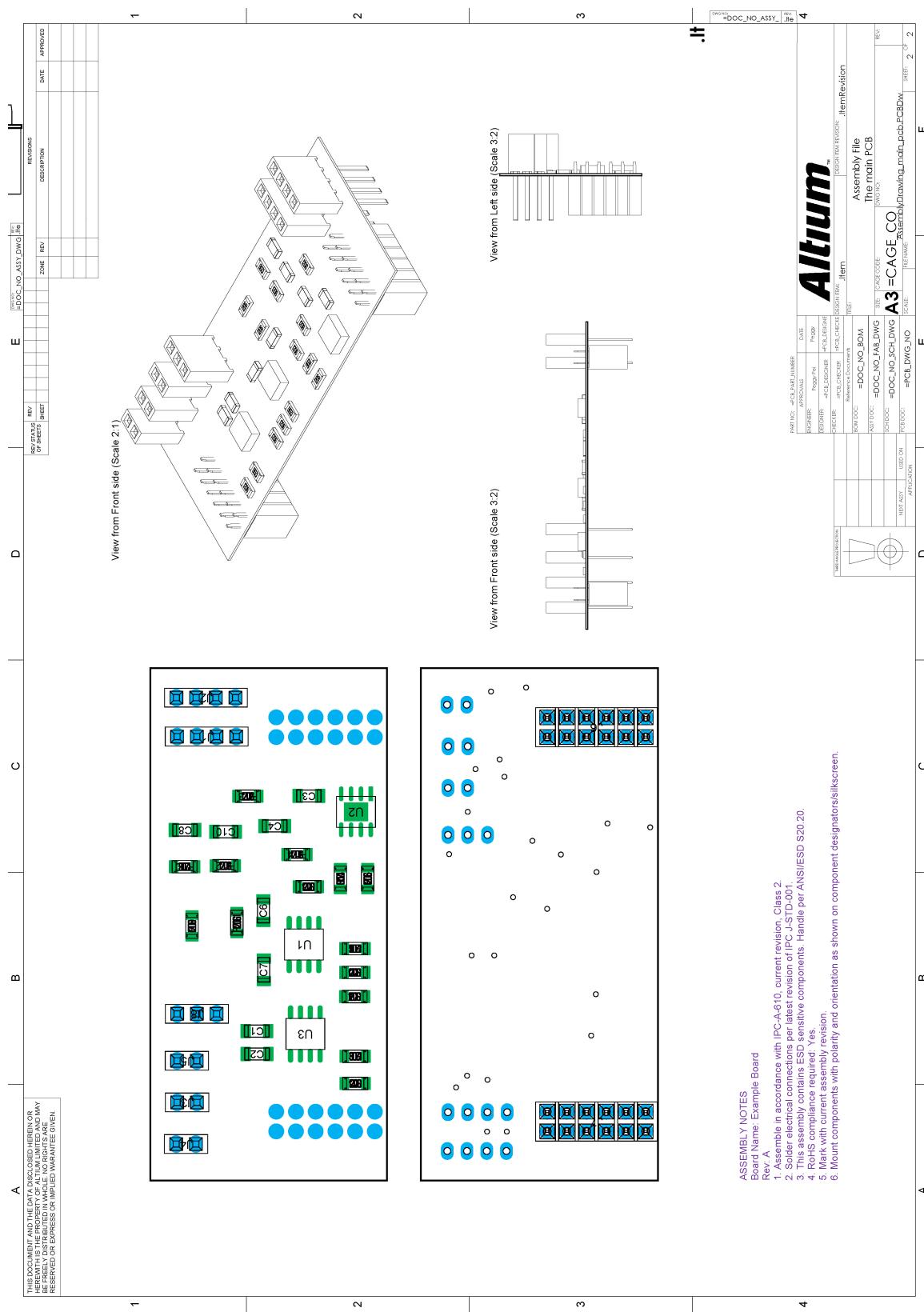
PCB Assembly and Fabrication Files

Assembly and fabrication files with detailed dimensions and notes for electronics outputs, the main PCB and the temperature sensing PCB, are attached and shown in figure B.1a and B.2a, respectively.



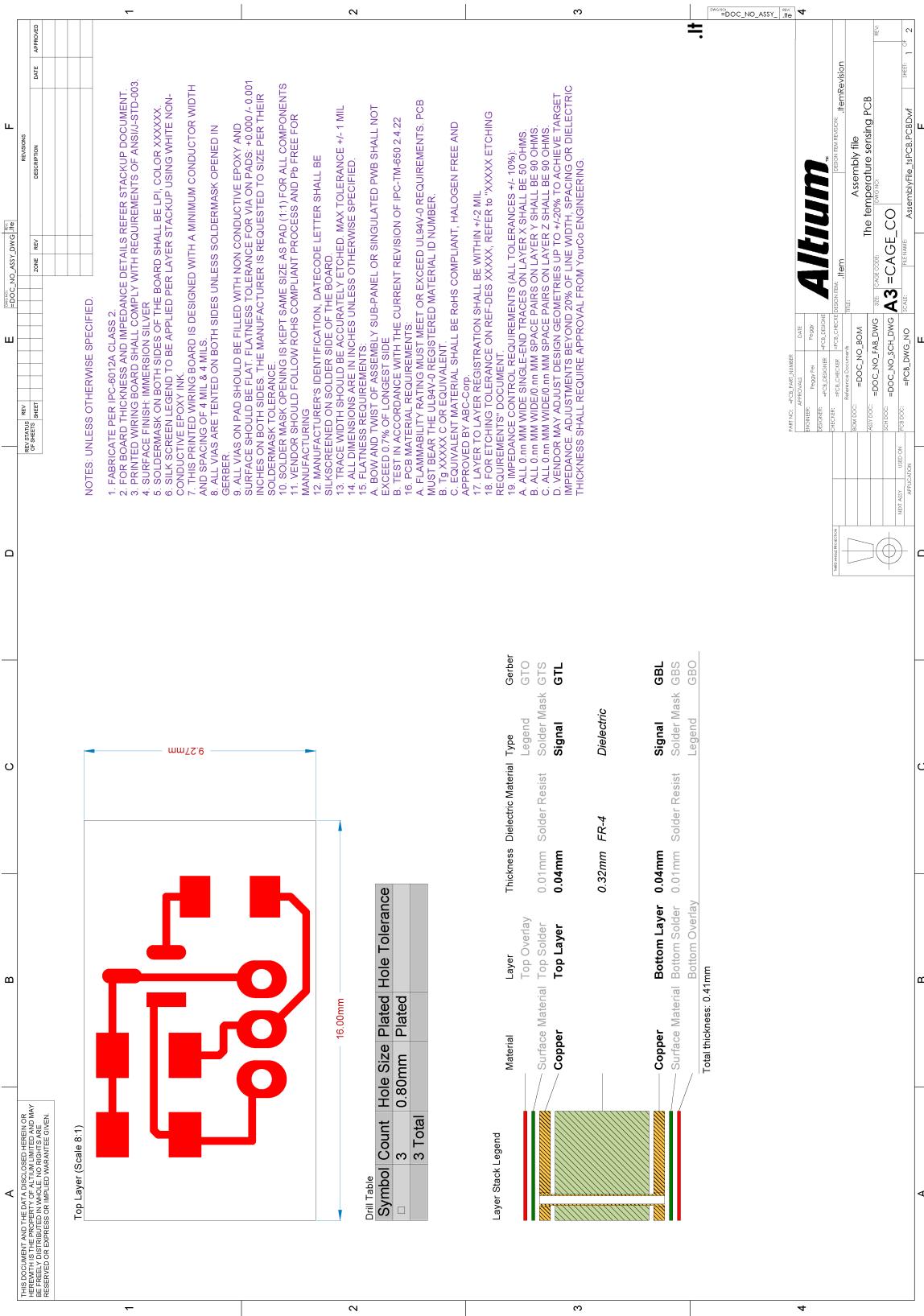
(a) Page 1/2

Figure B.1: Assembly and fabrication draft for the main PCB



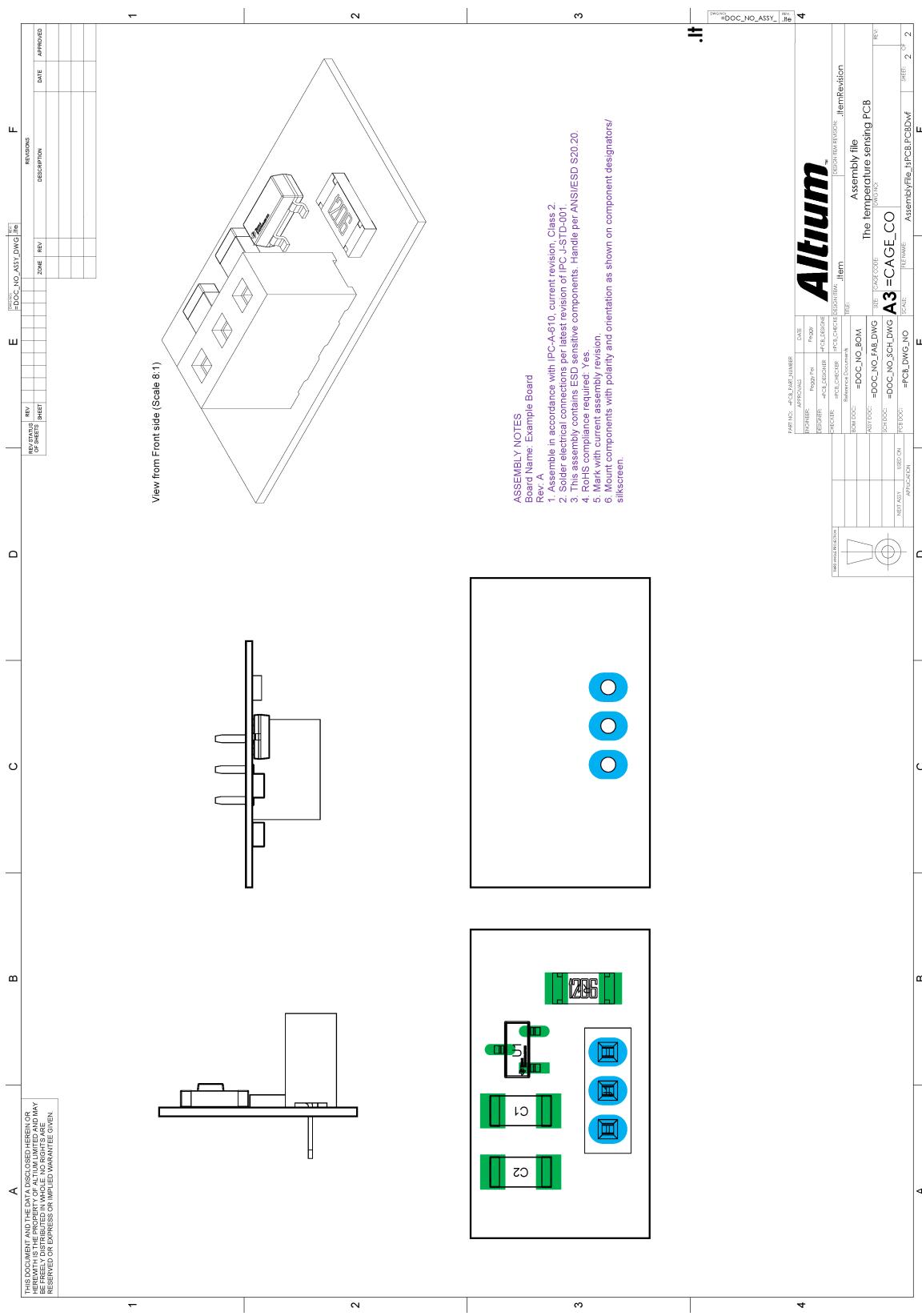
(b) Page 2/2

Figure B.1: Assembly and fabrication draft for the main PCB(cont)



(a) Page 1/2

Figure B.2: Assembly and fabrication draft for the temperature sensing PCB



(b) Page 2/2

Figure B.2: Assembly and fabrication draft for the temperature sensing PCB(cont)

Appendix C

Pinout view and System View

Table C.1 shows that the configuration of functional pins on **MCU**, which is summarised from the graphic configuration tool equipped on *STM32CubeIDE* as shown in figure C.1. Detailed parameter configuration of each pin can be found in the STM32 project uploaded on GitHub [7].

Parameter	PIN	Function	Specification
Tsensor	PF6	ADC3-IN4	Analogue reading pin for temperature sensor
Psensor	PC1	ADC1-IN11	Analogue reading pin for pressure sensor
Qsensor	PC3	ADC2-IN13	Analogue reading pin for pneumotachograph
EN-HP	PA1	GPIO	GPIO output to generate enable signal for the heating pad
EN-LSA	PA5	GPIO	GPIO output to generate actuating signals

Table C.1: Functional pin configuration OF MCU

The input and output of each subsystem are summarised in table C.2, for easy of final integration. The system view from the micro-processor side is shown in figure C.2, providing another perspective for system design on *STM32CubeIDE*.

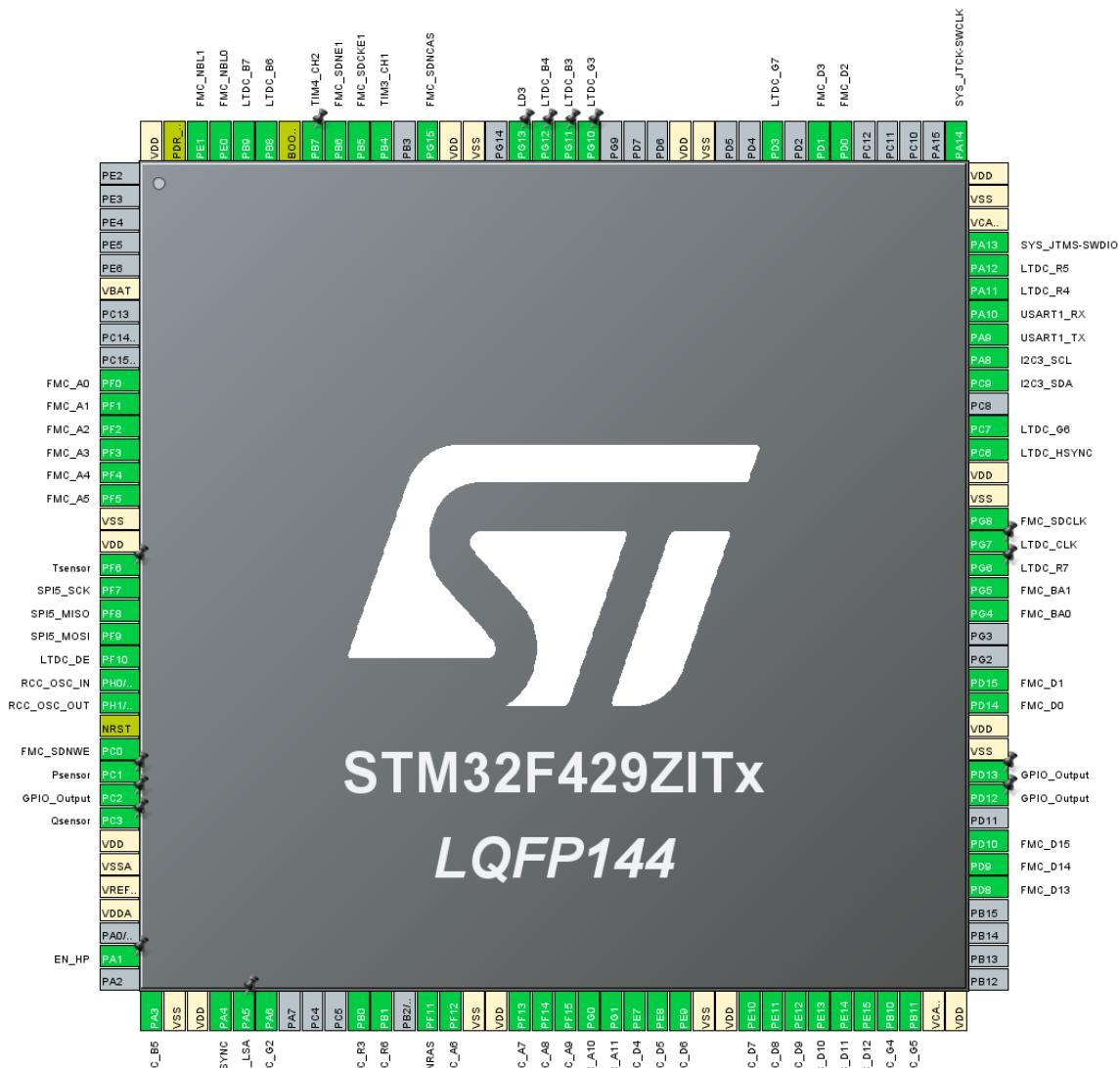
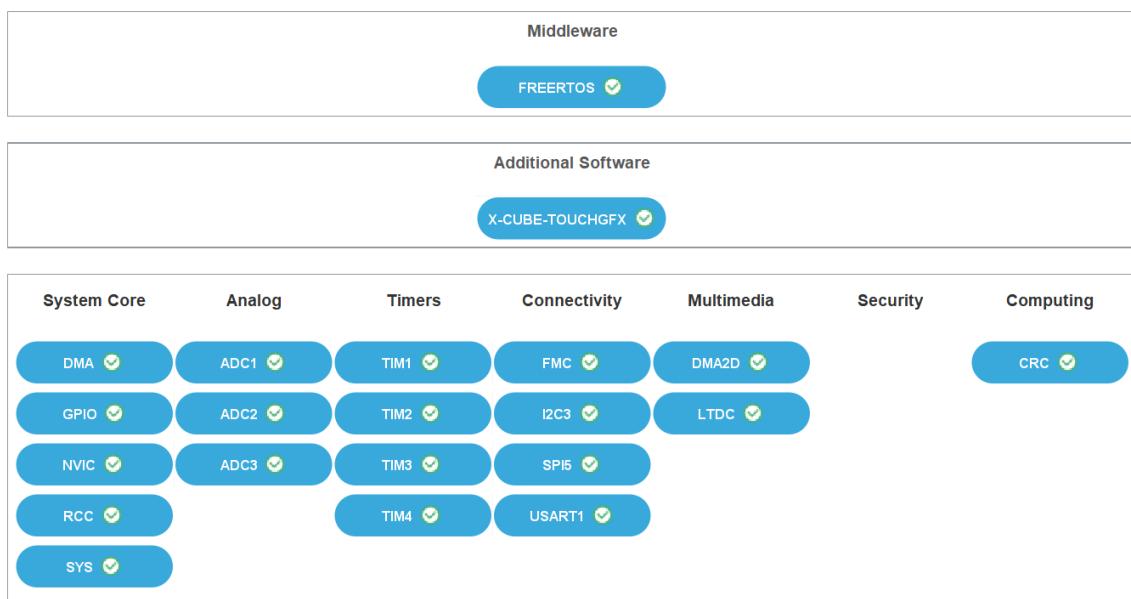


Figure C.1: Pinout view on *STM32CubeIDE*

Subsystem	Input	Output
Sensor system	VDD_PS OCS_PS GND VDD_FS OCS_FS VIN_t	V_P0 V_F0 VOUT_t
Power system	12V GND	3V3 GND
Heating system	EN_HP GND	V_HP GND
Actuating system	EN_LSA GND	V_LSA GND
MCU	3V3 OCS_PS V_P V_F GND	EN_HP V_F0 EN_LSA

Table C.2: System I&O

Figure C.2: System view on *STM32CubeIDE*

Appendix D

Cryptic configuration of the device

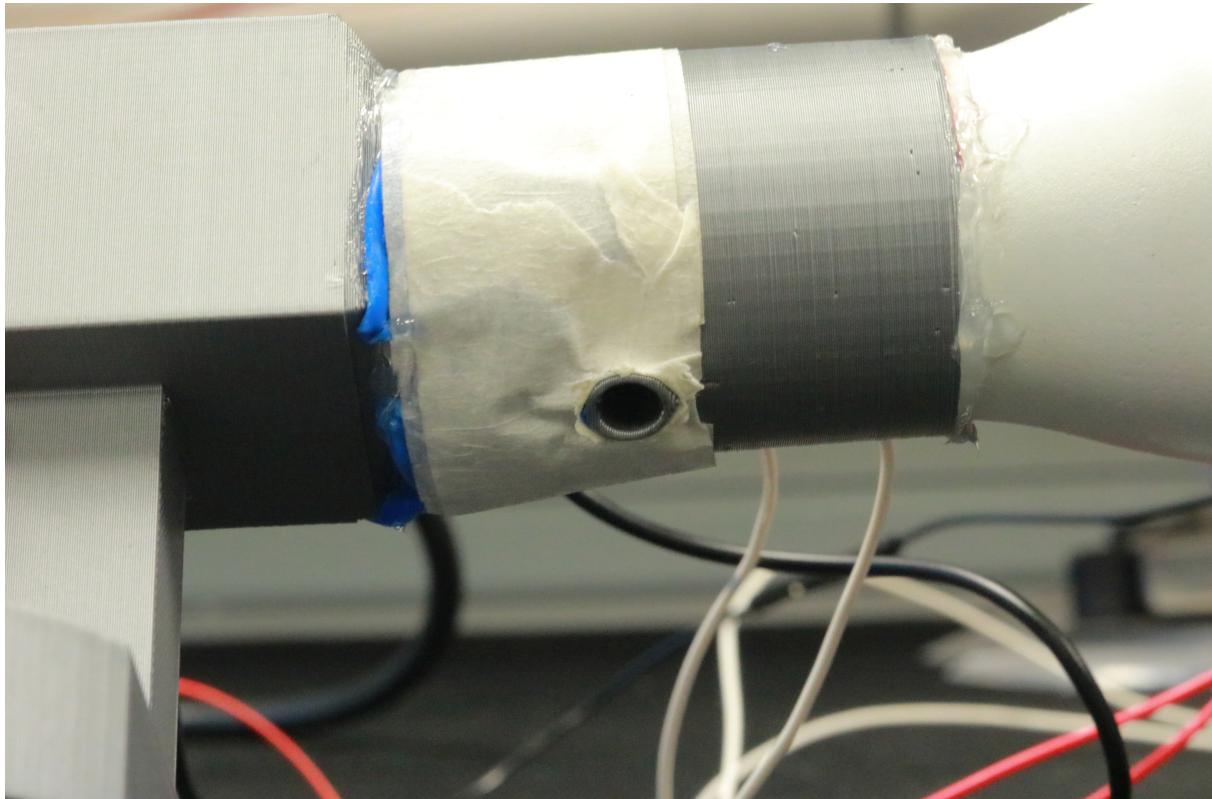


Figure D.1: Air outlet on the bottom of membrane connector

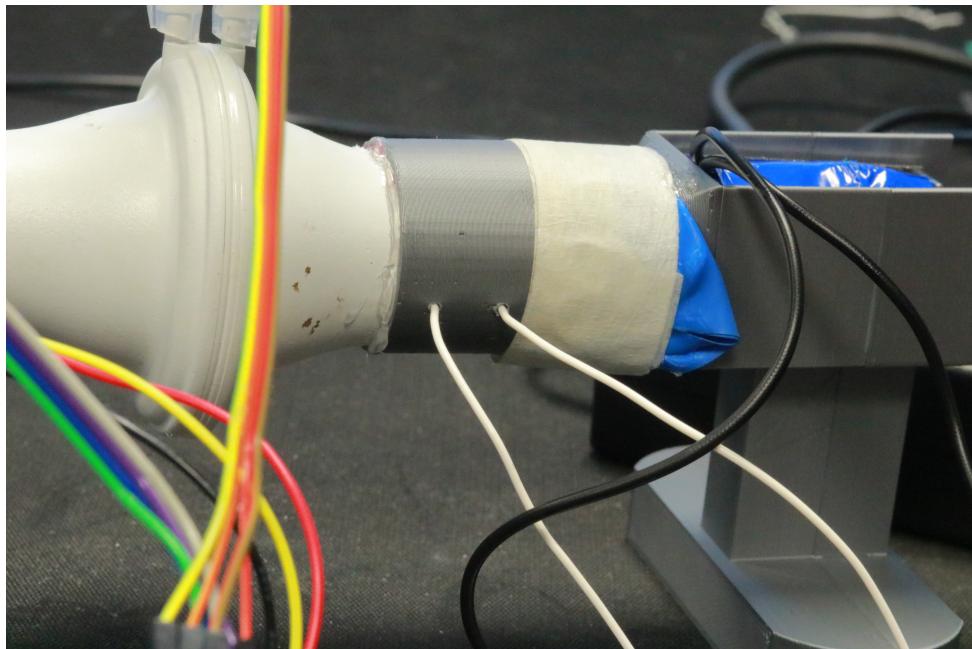


Figure D.2: Heating pad position on the back view

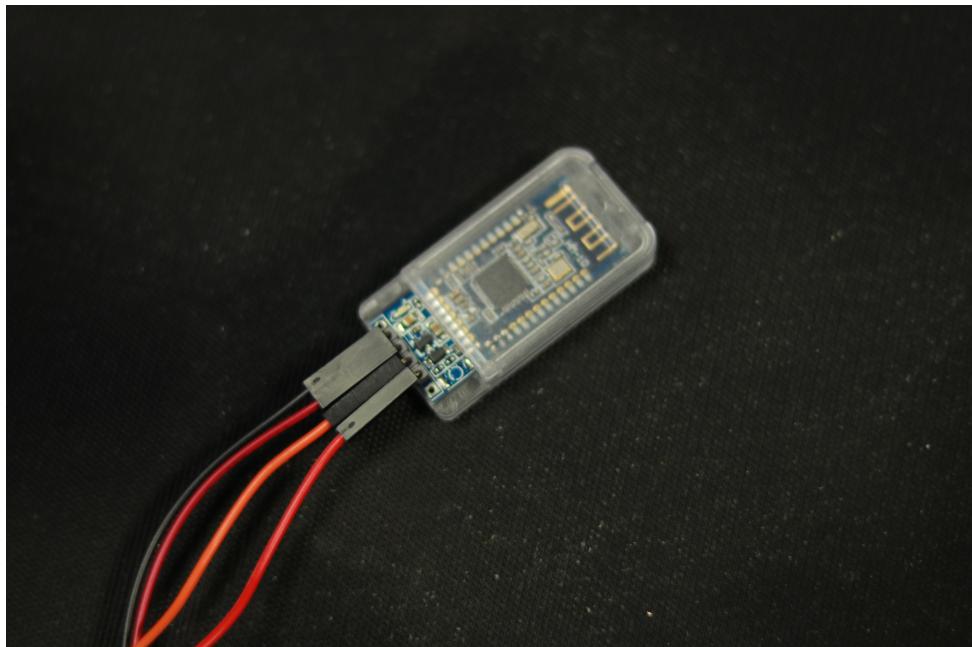


Figure D.3: HM-10 as the BLE module