



Universidade de Brasília

Instituto de Ciências Exatas
Faculdade de Tecnologia

Desenvolvimento de um ventilador oscilatório de alta frequência

Iago Cardoso Cortes

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador

Prof. Dr. Adson Ferreira da Rocha

Brasília
2019

Dedicatória

Dedico este trabalho a todos que me auxiliaram direta ou indiretamente para a sua conclusão.

Agradecimentos

Agradeço a meus pais e minha irmã, por sempre acreditarem em mim e por todo o suporte ao longo da minha vida.

Aos meus amigos, em especial ao Paulo pela paciência.

A todos os colegas com os quais convivi ao longo da graduação.

Ao professor Adson por sempre se disponibilizar para ajudar quando necessário.

Resumo

Muitas vezes pacientes que dependem do auxílio de aparelhos para respiração necessitam de alternativas à ventilação mecânica convencional, pois essa pode ser muito danosa aos seus pulmões. O ventilador oscilatório pulmonar de alta frequência é uma alternativa mais viável e segura para esses casos, para tanto, é necessária a garantia de seu correto funcionamento. Tendo isto em vista foi proposto o desenvolvimento de três módulos compostos por *Arduinos Pro Mini* e os circuitos eletrônicos necessários para a interação entre o microprocessador e o dispositivo de auxílio de respiração. Foram desenvolvidos um gerador de sinais simétricos senoidais para alimentar e definir o comportamento do ventilador; um módulo que foi conectado à saída do ventilador, que é responsável pela averiguação de que a pressão do ar fornecido ao paciente se mantém contínua em diferentes frequências e pontos; um sensor usado para monitorar o fluxo de ar pelo tubo.

Palavras-chave: Ventilação pulmonar oscilatória de alta frequência, Arduino, I2C

Abstract

Often patients who rely on the aid of breathing apparatus need alternatives to conventional mechanical ventilation, as it can be very damaging to their lungs. The high-frequency pulmonary oscillatory ventilator is a more viable and safe alternative for those cases, for that to happen it is necessary to guarantee its correct functioning. With this in mind, the development of three modules composed by Arduinos Pro Mini and the electronic circuits necessary for the interaction between the microprocessor and the breathing aid device was proposed. A symmetrical sinusoidal signal generator was developed to feed and define the ventilator behavior; one module that was connected to the ventilator outlet , which is responsible for ascertaining that the air pressure supplied to the patient is kept continuous at different frequencies and points; a sensor used to monitor the flow of air through the tube.

Keywords: High-frequency oscillatory pulmonary ventilation, Arduino, I2C

Sumário

1	Introdução	1
1.1	Ambientação	1
1.2	Objetivos do projeto	1
1.3	Proposta do projeto	2
1.4	Metodologia	2
1.5	Organização do projeto	3
2	Revisão Bibliográfica	4
2.1	Ventilação artificial	4
2.1.1	Ventilador mecânico atualmente	5
2.1.2	Ventilador mecânico convencional	6
2.1.3	Ventilador oscilatório de alta frequência	7
2.1.4	Divergências entre <i>CMVs</i> e <i>HFOVs</i>	8
2.2	MCP4725	8
2.3	SFM3000	9
2.4	MPX2010	10
2.5	Arduino	11
2.5.1	Interrupções	11
2.5.2	Protocolo <i>I2C</i>	12
3	Projeto	14
3.1	Fontes simétricas	14
3.2	Gerador de sinais senoidais	16
3.2.1	Como programar um <i>Arduino Pro Mini</i>	16
3.2.2	O <i>software</i>	16
3.2.3	O circuito somador e o amplificador inversor	18
3.3	Sensor de fluxo	18
3.4	Sensor de pressão	20
3.4.1	A calibração	20

3.4.2 O <i>software</i>	21
3.5 A montagem do ventilador	22
3.6 A coleta de dados	23
4 Resultados	27
4.1 Análise dos gráficos	27
5 Conclusão	34
5.1 Trabalhos futuros	34
Referências	36

Lista de Figuras

1.1	Robôs cirúrgicos.	2
2.1	Pulmotor.	5
2.2	Pulmão de aço.	6
2.3	Ilustração de um <i>HFOV</i>	7
2.4	MCP4725.	9
2.5	SFM3000.	10
2.6	MPX2010.	10
2.7	Arduino Pro Mini.	11
2.8	Diagrama da execução de uma interrupção.	12
2.9	Barramento <i>I2C</i>	13
3.1	Esquemático da fonte simétrica de 5V.	15
3.2	Fontes simétricas de 5 e 12V montadas.	15
3.3	Esquemático do circuito somador e do amplificador inversor.	18
3.4	Gerador de sistemas senoidais montado.	19
3.5	Fluxograma do gerador de sinais senoidais.	19
3.6	Esquemático do circuito montado para amplificar a diferença das leituras do sensor de pressão.	21
3.7	Circuito do sensor de pressão montado.	21
3.8	Calibração do circuito do sensor de pressão.	22
3.9	Esquemático do projeto.	23
3.10	Projeto montado.	24
4.1	Gráfico dos dados dos sensores de fluxo e de pressão à 5Hz e 1V de amplitude.	27
4.2	Gráfico dos dados dos sensores de fluxo e de pressão à 9Hz e 1V de amplitude.	28
4.3	Gráfico dos dados dos sensores de fluxo e de pressão à 12Hz e 1V de amplitude.	28
4.4	Gráfico dos dados dos sensores de fluxo e de pressão à 15Hz e 1V de amplitude.	29
4.5	Gráfico dos dados dos sensores de fluxo e de pressão à 5Hz e 3V de amplitude.	30
4.6	Gráfico dos dados dos sensores de fluxo e de pressão à 9Hz e 3V de amplitude.	30

4.7	Gráfico dos dados dos sensores de fluxo e de pressão à 12Hz e 3V de amplitude.	31
4.8	Gráfico dos dados dos sensores de fluxo e de pressão à 15Hz e 3V de amplitude.	31
4.9	Gráfico dos dados dos sensores de fluxo e de pressão à 5Hz e 5V de amplitude.	32
4.10	Gráfico dos dados dos sensores de fluxo e de pressão à 9Hz e 5V de amplitude.	32
4.11	Gráfico dos dados dos sensores de fluxo e de pressão à 12Hz e 5V de amplitude.	33
4.12	Gráfico dos dados dos sensores de fluxo e de pressão à 15Hz e 5V de amplitude.	33

Lista de Abreviaturas e Siglas

ADC Analog-to-Digital Converter.

CI Circuito Integrado.

CMVs Conventional mechanical ventilators.

DAC Digital-to-Analog Converter.

EEPROM Electrically Erasable Programmable Read-Only Memory.

HFOV High Frequency Oscillatory Ventilation.

I2C Inter-Integrated Circuit.

IDE Integrated development environment.

ISR Interrupt Service Routine.

MAP Mean Airway Pressure.

PaCO₂ Pressão arterial de CO₂.

SCL Serial Clock.

SDA Serial Data.

SLM Standard Litre per Minute.

Capítulo 1

Introdução

Este capítulo define a ambientação, o escopo do projeto, a metodologia usada no desenvolvimento dele bem como uma síntese do o que será apresentado nos demais capítulos.

1.1 Ambientação

No mundo industrializado a inovação tecnológica vem acontecendo de forma tão rápida que ela já faz parte de quase todos os aspectos da vida do ser humano. Uma das áreas de maior destaque quanto ao uso de tecnologia em seu meio é a medicina. Esta, por mais que tenha uma grande história, só passou a ter o uso de mecanismos tecnológicos com as mais diversas aplicações recentemente [1].

Com a integração de engenheiros no meio da medicina a disciplina de engenharia biomédica surgiu, e esta desenvolveu diversas ferramentas que assistem nos tratamentos dos mais diversos tipos de problemas voltados à saúde em múltiplas etapas desse. Estes engenheiros aplicam princípios de diversas áreas de conhecimento de engenharia para entender, modificar ou controlar sistemas biológicos [1].

A área de engenharia biomédica contribuiu no desenvolvimento de vários conceitos revolucionários e salvadores, alguns destes conceitos são: órgãos artificiais, robôs cirúrgicos (Figura 1.1), próteses avançadas, novos medicamentos farmacêuticos, diálise renal, etc [2].

1.2 Objetivos do projeto

Dado um ventilador oscilatório pulmonar de alta frequência pré-existente, composto por um alto falante, um diafragma, um tubo endotraqueal e um compressor de ar, o projeto consiste em:



Figura 1.1: Robôs cirúrgicos.

1. Desenvolver um gerador de sinais simétricos senoidais com amplitude e frequência variáveis para alimentar o amplificador que é conectado ao alto-falante e definir, portanto, sua frequência de operação.
2. Integrar ao tubo endotraqueal do aparelho de auxílio respiratório dois sensores: um sensor de pressão e um de fluxo.
3. Programar os módulos anteriores de forma que eles possam ser iniciados e controlados por um módulo controlador externo e que este receba dados de operação dos diversos módulos em operação.
4. Com todo o sistema montado realizar testes e validar a viabilidade do projeto.

1.3 Proposta do projeto

A proposta do projeto é, com todo o *High Frequency Oscillatory Ventilation (HFOV)* montado, e os módulos individuais funcionando como esperado, testar em bancada as funcionalidades do protótipo, avaliar seu potencial para futuro uso, e permitir a comunicação com um módulo controlador externo.

1.4 Metodologia

Para o desenvolvimento deste trabalho foi realizada uma pesquisa primeiramente exploratória na qual foram utilizadas fontes primárias para melhor entender a ambientação do assunto a ser tratado (engenharia biomédica). Foi necessário então entender a origem da ventilação artificial e como ela evoluiu ao longo do tempo para, finalmente, chegar a como ela se encontra hoje.

A pesquisa então se voltou para a leitura de diversos artigos científicos, livros e patentes para entender como ocorre o funcionamento e como são construídos os ventiladores pulmonares e quais seus critérios de uso. Neste ponto a pesquisa foi focada em ventiladores oscilatórios de alta frequência, visto que este é o tema do projeto.

Com estes conhecimentos foi possível compreender as propostas do projeto, bem como elas se interagem com um ventilador e, então, os módulos propostos nos objetivos foram construídos.

1.5 Organização do projeto

O capítulo 2 apresenta uma revisão teórica de ventiladores pulmonares de alta frequência e dos dispositivos usados no projeto. No capítulo 3 cada etapa do projeto é descrita. No capítulo 4 os resultados dos testes do projeto são apresentados e discutidos. O capítulo 5 contém a conclusão baseada nos resultados obtidos no capítulo 4.

Capítulo 2

Revisão Bibliográfica

Este capítulo descreve os conceitos teóricos necessários para o desenvolvimento do projeto, bem como as principais ferramentas utilizadas no *HFOV*. O capítulo está estruturado da seguinte forma: a seção 2.1 aborda o *HFOV*; as seções 2.2, 2.3 e 2.4 descrevem, respectivamente, a finalidade dos componentes: MCP4725, MPX2010, SFM3100. Na seção 2.5 é realizada uma introdução ao Arduino, ao microcontrolador *Arduino Pro Mini* e a duas das tecnologias presentes no arduino que foram usadas no projeto, interrupções e protocolo *I2C*.

2.1 Ventilação artificial

Constitui ventilação artificial qualquer meio, sem participação do sistema respiratório do paciente, de realizar a troca de ar entre o pulmão do paciente e o ar ao seu redor [3]. É possível observar ao longo da história diversos experimentos que podem ser caracterizados como tentativas de fazer exatamente isso. O médico, cirurgião e filósofo grego Aelius Galenus (129 AD - 210 AD), em um de seus experimentos, operou um fole para inflar o pulmão de um animal morto. No ano 1664, o mesmo experimento foi realizado em um cachorro, vivo, com peito aberto. Foi inserido um tubo ligado a um fole em sua traquéia e foi possível observar que a contínua operação do fole mantinha o coração do animal batendo e que, caso o fole fosse retirado, o coração apresentava movimentos convulsivos [4].

Com o passar do tempo, e com as novas descobertas realizadas, a forma como a ventilação artificial era realizada foi mudando. O uso da respiração boca a boca tornou-se uma forma bastante popular de ventilação artificial devido a três fatores: a facilidade de realização, o calor e a umidade presente no sopro do ressuscitador e a pressão do sopro que é suportada pelo pulmão do paciente. Com a descoberta do dióxido de carbono e do oxigênio, este método se tornou antiquado, visto que foi descoberto que a exalação de ar

é rica em dióxido de carbono [4]. Essas e outras descobertas ajudaram os cientistas da época a melhor entender o funcionamento do sistema respiratório humano e, portanto, a desenvolver melhores formas de realizar a ventilação pulmonar.

Por volta dos anos 1900, foi desenvolvido o pulmoter (Figura 2.1), um ventilador mecânico que usava um cilindro de oxigênio com um regulador e um medidor de pressão. O pulmoter era ligado a uma válvula controlada por um relógio responsável por acionar a liberação de ar a uma frequência de, aproximadamente, 0,2Hz para uma máscara. Para a fase de inspiração, este aparelho era colocado no paciente e a fase de expiração ocorria normalmente após a retirada do dispositivo [5].

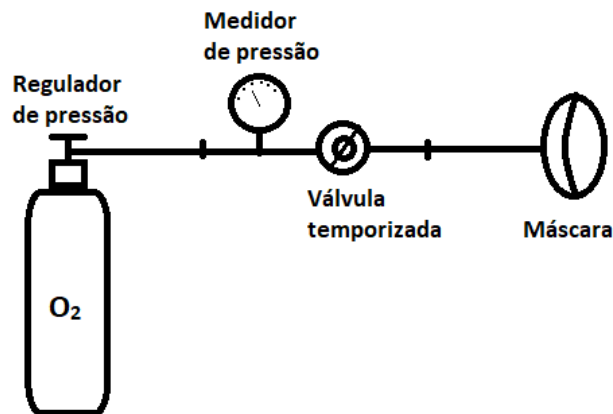


Figura 2.1: Pulmoter.

Alguns anos depois, o primeiro pulmão de aço (Figura 2.2), também conhecido como ventilador de pressão negativa, o qual fora amplamente utilizado, foi desenvolvido. Ele consistia em uma câmara hermeticamente fechada, em que o paciente era colocado, ficando apenas sua cabeça de fora, e era criada pressão negativa dentro do compartimento. Esse mecanismo fazia com que a caixa torácica do paciente se expandisse, o que permitia que o pulmão aspirasse ar pelas vias respiratórias do enfermo. Tal dispositivo, então, retornava a câmara para a pressão atmosférica e o ar do pulmão era expelido [5].

2.1.1 Ventilador mecânico atualmente

Atualmente, quando o enfermo manifesta graves problemas respiratórios, ou mesmo durante procedimentos cirúrgicos, pode ser necessário o auxílio de um ventilador pulmonar. A grande maioria dos ventiladores pulmonares utilizados em unidades de tratamento intensivo utilizam tecnologias similares. Esses são constituídos por um reservatório de ar pressurizado e uma unidade de controle responsável por controlar a válvula inspiratória. Para garantir o correto funcionamento do ventilador são utilizados sensores de pressão e

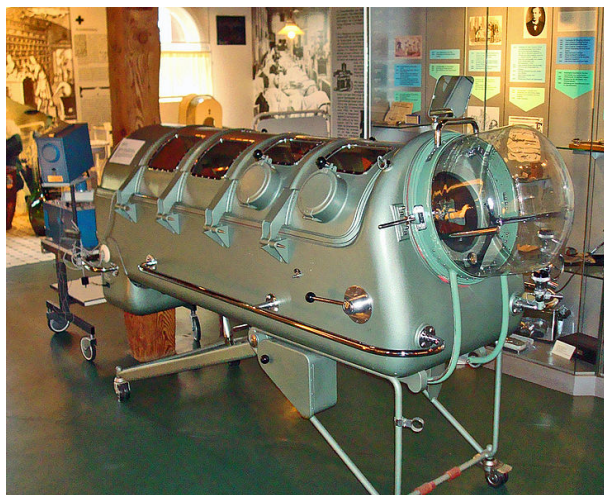


Figura 2.2: Pulmão de aço (Fonte: [6]).

de fluxo, que monitoram vazamentos de ar ou se o aparelho foi desconectado do paciente. [7].

O paciente é entubado e o tubo endotraqueal é conectado ao ventilador a fim de fornecer ar oxigenado para os pulmões e permitir a saída de dióxido de carbono [8]. O paciente pode ser monitorado por sensores de pressão e de fluxo, bem como pela sua frequência cardíaca, taxa de respiração, pressão sanguínea, saturação de oxigênio, entre outros [9].

2.1.2 Ventilador mecânico convencional

Existem três tipos de *Conventional mechanical ventilators (CMVs)*, quais sejam os com ciclo baseado em volume (mais usado), em pressão e em tempo (menos usado) [10]:

- Ventilador com ciclo baseado em volume: este ventilador controla as fases inspiratórias e expiratórias baseado no volume corrente, o volume normal de ar deslocado entre as fases respiratórias acima, de cada paciente. Este valor é, de aproximadamente, 7 mL/kg de peso corporal [11].
- Ventilador com ciclo baseado em pressão: muda entre as fases respiratórias quando o pulmão atinge uma pressão previamente estabelecida.
- Ventilador com ciclo baseado em tempo: faz a troca entre as fases respiratória após um tempo predeterminado.

2.1.3 Ventilador oscilatório de alta frequência

Estes ventiladores, diferentemente dos outros ventiladores mecânicos, incorpora uma fase de expiração ativa, isto é, a expiração passa a ser realizada pelo próprio dispositivo. Logo, quando os pulmões do paciente, seja ele adulto, criança ou, especialmente, recém-nascido prematuro, não são capazes de realizar a respiração de forma apropriada, em particular, na fase de expiração, este ventilador torna-se a escolha ideal [12].

Este ventilador, então, é usado em pacientes com lesões pulmonares agudas e síndrome de insuficiência respiratória aguda, visto que ele ventila o pulmão de forma estratégica com o objetivo de mitigar as lesões pulmonares induzidas pelo ventilador [13].

HFOVs, em geral, são compostos por um tubo ligado, em um lado, às vias aéreas do paciente por meio, por exemplo, de um tubo endotraqueal e, no outro, a um oscilador. Este dispositivo gera oscilação de pressão (de positiva para negativa e vice-versa) por meio de um pistão que move um diafragma plástico à uma frequência selecionada [12].

O pistão do oscilador, similar a um alto-falante, vibra o gás pressurizado a uma alta-frequência, de forma que a variação de pressão causada pela movimentação do pistão retira e repõe gás ao longo do tubo. Esse gás entra e sai pelas vias respiratórias do paciente e, assim, realiza a ventilação. Uma fonte de gás intercepta o tubo em que ocorrem as oscilações e o ar por essa fonte liberado flui ao longo do tubo de forma constante e contínua de modo que, com as variações de pressão, em uma fase da oscilação, ar fresco seja fornecido ao paciente e, na outra fase, ar rico em CO_2 é extraído. [14]. Este mecanismo foi ilustrado na Figura 2.3.

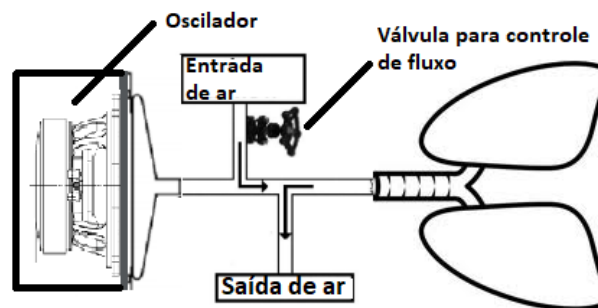


Figura 2.3: Ilustração de um *HFOV*.

Esse fluxo constante de ar também é importante pois mantém os pulmões e alvéolos abertos por meio de uma pressão positiva média nas vias aéreas sobre a qual ocorrem as oscilações, o que permite o uso de pequenos volumes correntes para oxigenar o paciente e remover o acúmulo de CO_2 nas vias aéreas. Detalhe fundamental para que seja realizada oxigenação efetiva [14].

A frequência de vibração do pistão em *HFOVs* costuma estar entre 3Hz e 15Hz, isto é, promove de 180 a 900 respirações por minuto, um número consideravelmente maior que a taxa de respiração normal de um ser-humano, 12 a 20 respirações por minuto [12]. Esta alta frequência significa que o volume corrente de ar fornecido para o pulmão seja bastante reduzido (até um décimo do volume corrente fornecido por um *CMV* [15]). O volume corrente provido pode ser controlado com a variação da amplitude do sinal que alimenta o pistão, quanto maior a amplitude de movimento e, conseqüentemente, da alimentação do pistão, maior a quantidade de volume corrente disponibilizado. O que acarreta em uma maior inflação e deflação do pulmão do paciente sobre a pressão positiva média nas suas vias aéreas [12].

A amplitude do movimento do pistão também é usada para o controle da Pressão arterial de CO₂ (PaCO₂), visto que, como foi discutido acima, o aumento da amplitude aumenta o volume corrente fornecido ao paciente e, conseqüentemente, reduz a PaCO₂ [16].

2.1.4 Divergências entre *CMVs* e *HFOVs*

São perceptíveis algumas características que distinguem os ventiladores convencionais dos oscilatórios de alta-frequência.

- Nos *HFOVs* é possível controlar de forma independente tanto a ventilação quanto a oxigenação [17].
- Os *CMVs* só ventilam durante a fase de inspiração, logo a expiração ocorre naturalmente como reação à ventilação [12], enquanto que os *HFOVs* possuem fase de expiração ativa.
- Os *CMVs* fornecem ar respirável a um volume e frequência próximos aos de uma pessoa respirando espontaneamente [18] [14]. *HFOVs* provêm pequenos volumes correntes de ar a frequências muitas vezes mais altas que a frequência de respiração humana normal.
- *HFOVs* mantém a abertura dos pulmões por meio de uma constante pressão média nas vias aéreas do paciente, assim realizando a oxigenação de forma efetiva sem o colapso dos alvéolos pulmonares [19].

2.2 MCP4725

O MCP4725 é um conversor de sinais digitais, de 12 *bits*, para analógicos de alta precisão e baixo consumo, no qual a tensão de operação está entre 2,7V e 5,5V. A comunicação

com este conversor é realizada através de uma interface *I2C* capaz de suportar as três classes de velocidade de operação de 100kbps, 400kbps ou 3,4Mbps [20].

Este *Digital-to-Analog Converter (DAC)* possui memória não volátil, *Electrically Erasable Programmable Read-Only Memory (EEPROM)*, que pode ser usada para salvar um valor de entrada a ser convertido ou mesmo armazenar os *bits* de configuração, o que permite que o valor de saída desejado esteja imediatamente disponível após este dispositivo ser ligado [20].

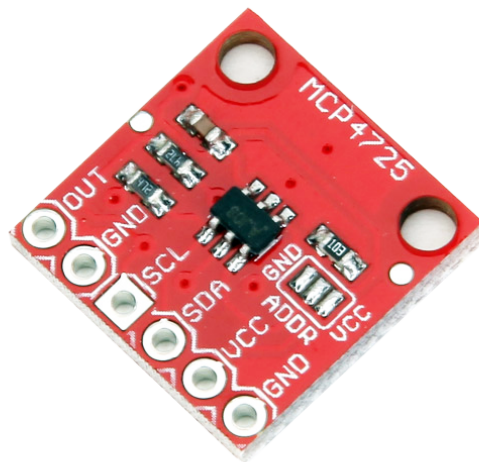


Figura 2.4: MCP4725 (Fonte: [21]).

2.3 SFM3000

O SFM3000 é um sensor digital bidirecional de fluxo que mede a taxa de vazão de oxigênio e outros gases não agressivos. Sua acurácia (1.5% do valor medido), rápida taxa de atualização (5ms) e a baixa perda de pressão do ar ao longo do canal de vazão fazem desse sensor uma escolha adequada para aplicações médicas como a ventilação pulmonar. Este sensor é alimentado por uma fonte de 5V e a comunicação com esse é realizada por meio de uma interface *I2C* [22].

O valor medido pelo sensor pode ser convertido para *Standard Litre per Minute (SLM)* usando a Equação 2.1. Nesta o valor medido é referente à leitura da saída do sensor; o valor de *offset* de fluxo do sensor é de 32000; e o denominador da equação é o fator de escala para o ar, N₂ e O₂ que possuem, respectivamente, os valores: 140, 140 e 142.8.

$$fluxo[slm] = \frac{valor\ medido - offset}{fator\ de\ escala} \quad (2.1)$$



Figura 2.5: SFM3000.

2.4 MPX2010

O MPX2010 é um sensor de pressão diferencial desenvolvido pela *NXP Semiconductors* que fornece como saída uma voltagem equivalente à pressão de entrada com alta precisão e linear. Esse sensor apresenta uma faixa de pressão diferencial de até 1.45psi e é ratiométrico à sua tensão de alimentação operacional de 3V a 6V.

Este sensor possui dois lados P1 e P2, a pressão diferencial é medida ao aplicar pressão sobre o lado P1 em relação ao lado do vácuo P2, logo quanto maior a pressão aplicada sobre P1 maior a tensão de saída. Similarmente, a saída aumenta à medida que é aumentado o vácuo em P2 em relação a P1. [23]



Figura 2.6: MPX2010.

2.5 Arduino

Arduino é uma plataforma física de computação vendida a um preço módico voltada tanto para o versado quanto para o leigo no uso de microcontroladores, na qual tanto o *hardware* quanto o *software* são *open source*. A comunidade Arduino é aberta e atuante, e conta com membros de diversos níveis de experiência que assistem outros membros em seus projetos e disponibilizam tutoriais em diversos tópicos [24].

A empresa Arduino comercializa placas de desenvolvimento baseadas em microcontroladores conhecidas como *Arduino Modules*, que são, mais comumente, programadas na linguagem C utilizando o *Arduino Integrated development environment (IDE)* e contando com a vasta coleção de bibliotecas disponíveis *online* [25].

No projeto foram utilizadas as placas de desenvolvimento *Arduino Pro Mini*, uma é exibida na Figura 2.7. Estas placas possuem preço baixo, são flexíveis e pequenas, o que a qualificam como uma boa escolha para uso em sistemas embarcados. O *Arduino Pro Mini* é um microcontrolador baseado no *ATMega328p* que roda a 5V e 16MHz [25]. Neste projeto duas tecnologias presentes neste Arduino foram: interrupções e protocolo *Inter-Integrated Circuit (I2C)*.

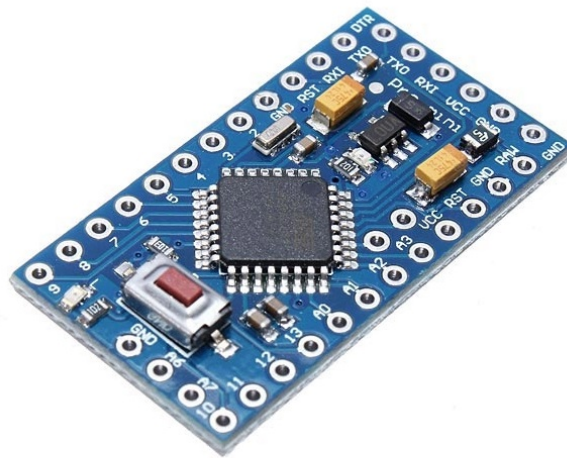


Figura 2.7: Arduino Pro Mini (Fonte: [26]).

2.5.1 Interrupções

A interrupção é uma forma de fazer com que um evento externo (e.g. clicar em um botão) ou uma instrução no *software* (e.g. *timer*) suspenda a execução do programa principal

para que uma outra rotina seja executada e após seu termino o processador volta a executar a rotina principal Figura 2.8. No Arduino isso é feito por meio da declaração `attachInterrupt` que possui três parâmetros [27]:

1. “`digitalPinToInterrupt(pin)`”: o pino de entrada digital que é usado para a interrupção, no Arduino Pro Mini os pinos 2 e 3 são usados para interrupções.
2. “`ISR`”: *Interrupt Service Routine (ISR)* é um evento que não possui parâmetros, não retorna nada, deve possuir uma pequena quantidade de instruções e ser executado em pouco tempo. São estas as instruções que são executadas quando ocorre a interrupção.
3. “`mode`”: tipo de transição que irá engatilhar a interrupção. Pode ser: “`RISING`” quando o pino vai de `LOW` para `HIGH`; “`FALLING`” quando ocorre o contrário; “`CHANGE`” quando o estado do pino muda.

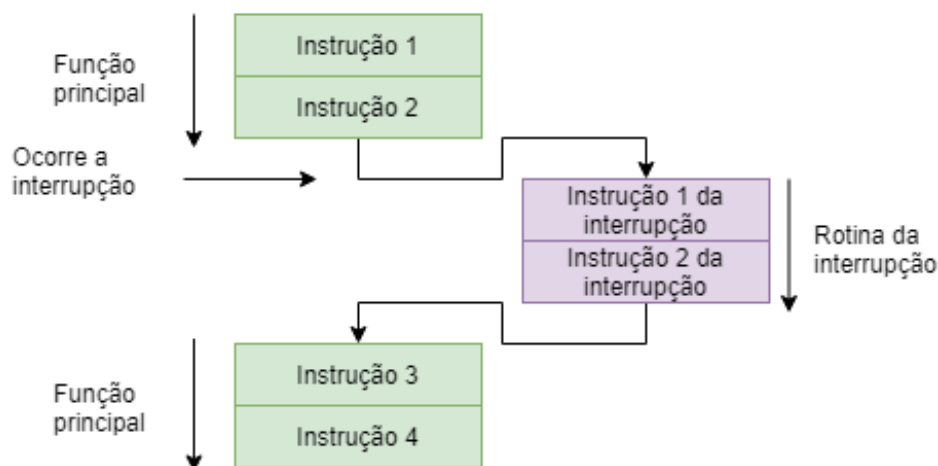


Figura 2.8: Diagrama da execução de uma interrupção.

2.5.2 Protocolo *I2C*

Este é um protocolo de comunicação síncrono (dispositivos em comunicação por este protocolo devem compartilhar o mesmo sinal de *clock*), no qual a comunicação ocorre com o uso de apenas dois fios bidirecionais *Serial Data (SDA)* e *Serial Clock (SCL)*, responsáveis, respectivamente, por transportar os dados e por sincronizar a transferência destes pelo barramento *I2C*. Os dispositivos que integram e compartilham este barramento são chamados ou de mestres ou de escravos [28].

A comunicação é iniciada pelo mestre que mantém *SCL* alto e faz *SDA* ir de alto para baixo, gerando assim a condição de '*START*' da comunicação, que é um aviso a todos os dispositivos conectados no barramento que eles devem entrar no modo de escuta. O mestre então envia um identificador único (endereço de 7 ou 10 *bits*) para referenciar o escravo com o qual ele deseja se comunicar, esse endereço vem seguido do *bit R/W* que define a direção da transmissão (caso seja alto o escravo irá transmitir, caso baixo o mestre) [28].

Caso algum escravo no barramento tenha o endereço igual ao enviado pelo mestre, este escravo irá enviar um *bit* de *Acknowledgement (ACK)*, o mestre então irá transmitir ou ler os dados pela linha *SDA* em sequências de 8 *bits* seguidos de um *ACK* ou *NACK*, até que a transferência seja concluída com a condição de parada, com *SCL* alto ocorre uma transição de baixo para alto em *SDA*.

Os Arduinos usados neste projeto possuem um pino para *SDA* e um pino para *SCL*, estes pinos são, respectivamente, os pinos analógicos 4 e 5.

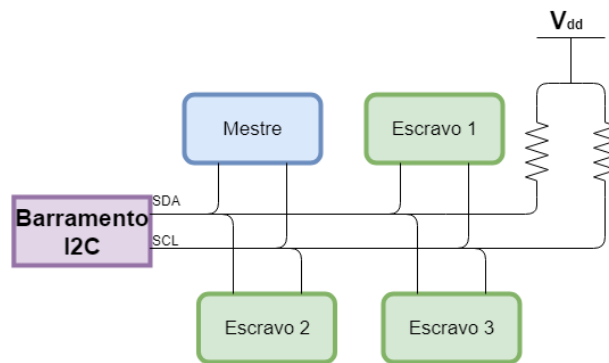


Figura 2.9: Barramento *I2C*.

Capítulo 3

Projeto

Este capítulo discute a preparação, o desenvolvimento e a montagem de cada módulo e do projeto em si. O capítulo está estruturado da seguinte forma: a seção 3.1 descreve as fontes simétricas de 5 e 12V; na seção 3.2 é apresentada a forma como o gerador de sinais senoidais foi desenvolvido. As seções 3.3 e 3.4 descrevem, respectivamente, os projetos dos sensores de fluxo e de pressão. Na sessão 3.5, é descrita a montagem do projeto por etapas e, por último, na sessão 3.6 a forma como foi realizada a coleta de dados e a posterior transformação destes dados em informação é relatada.

3.1 Fontes simétricas

Durante o levantamento de requisitos foi detectada a necessidade de uma fonte simétrica. Os *arduinos pro mini*, os sensores de pressão e de fluxo e o conversor *DAC* são alimentados por 5V, porém para a geração do sinal de corrente alternada senoidal foi também necessário o uso de uma fonte de -5V.

A fonte simétrica de 5V deve ser alimentada por um sinal de 7-0-7 VAC até 12-0-12 VAC ou +/- 9 até 18 VDC. Este sinal de entrada, quando AC, é convertido para DC por uma ponte retificadora, e os reguladores LM7805 e LM7905 foram utilizados para manter o nível de tensão constante em 5VDC e -5VDC respectivamente. Estes circuitos integrados oferecem proteção contra sobrecarga térmica e curtos-circuitos. O circuito também contém capacitores filtradores para reduzir a ondulação do sinal DC na saída. A fonte foi alimentada por um transformador abaixador (12V+12V/1A).

Similarmente, a fonte simétrica de 12V foi montada usando uma ponte retificadora na qual os componentes positivos e negativos do sinal de saída desta foram conectados a capacitores eletrolíticos de $1000\mu\text{F}$, e, no caso da parte positiva do sinal, ao regulador LM7812 e na parte negativa ao regulador LM7912. A fonte foi alimentada pelo mesmo transformador da fonte de 5VDC.

Os dois circuitos foram montados em uma placa de circuito impresso e colados, juntos ao transformador, em um suporte composto por uma placa de acrílico presa a uma base de eucatex por rebites. O esquemático do circuito da fonte de 5V pode ser visto na Figura 3.1 e os circuitos das duas fontes montadas na Figura 3.2.

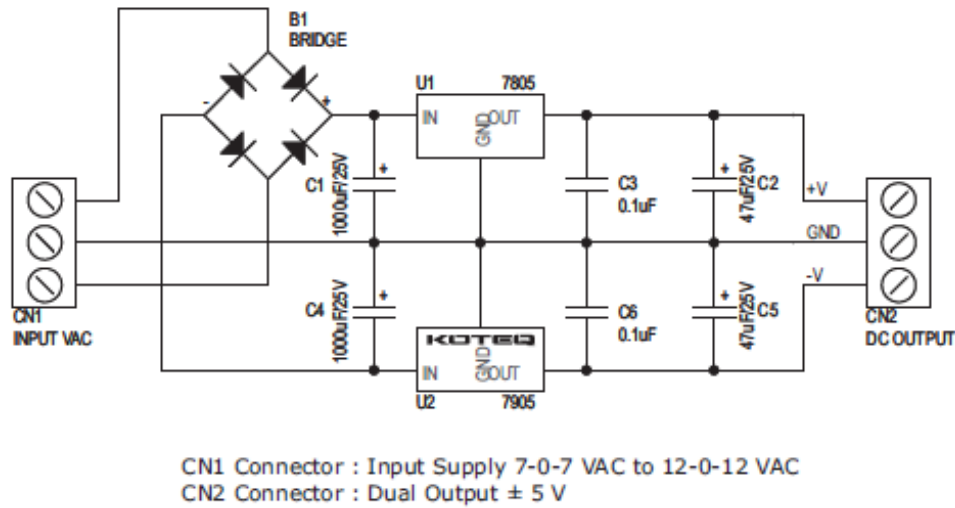


Figura 3.1: Esquemático da fonte simétrica de 5V (Fonte: [29]).

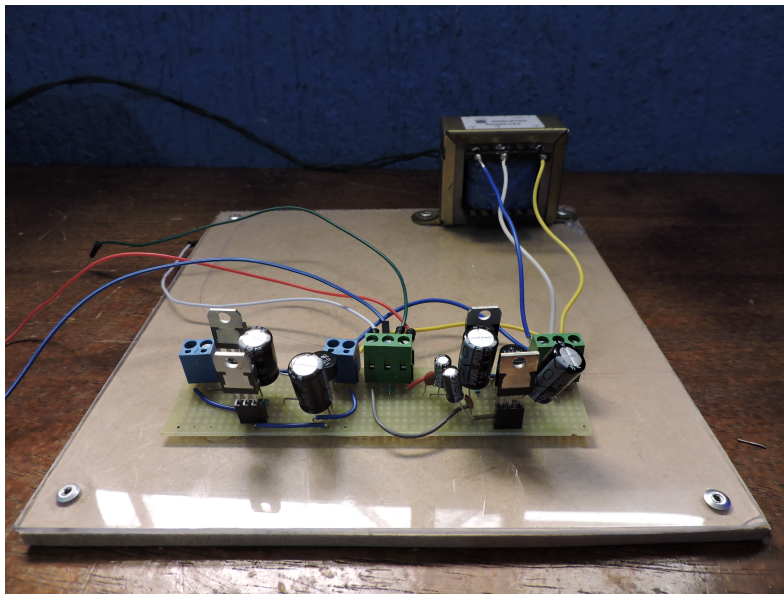


Figura 3.2: Fontes simétricas de 5 e 12V montadas.

3.2 Gerador de sinais senoidais

Para gerar o sinal de entrada do amplificador do alto-falante foi desenvolvido um gerador de sinais senoidais com amplitude e frequência selecionáveis.

3.2.1 Como programar um *Arduino Pro Mini*

Como o *Arduino Pro Mini* não possui entrada USB, ele foi programado usando um *Arduino Uno* de acordo com os seguintes passos:

1. Remova o chip ATmega328P do *Arduino Uno*;
2. conecte os pinos de reset, 5V, *GND*, *TX* e *RX* do *Pro Mini* nos mesmos pinos do *Uno*;
3. ligue o *Uno* ao computador com o cabo USB;
4. na opção “Placa” em ferramentas na IDE do Arduino selecione “*Arduino Pro or Pro Mini*”;
5. na opção “Processador” em ferramentas selecione “ATmega328P (5V, 16MHz)”;
6. na opção “Porta” em ferramentas escolha porta serial na qual o Arduino está conectado e clique para carregar o programa.

3.2.2 O *software*

O sinal senoidal foi gerado a partir de um programa em um *Arduino Pro Mini* baseado em um conjunto de amostras de valores de seno. Este programa recebe um valor para a amplitude de 0 a 4,2V, um de frequência de 0 a 50Hz e, então, produz uma onda senoidal que, devido às restrições de tensão de operação do *Arduino* (0 a 5V), é centrada em 2,5V.

As amostras de seno foram geradas a partir do script em python abaixo. Este código gera uma amostra de seno com três casas decimais a cada terceiro grau de um círculo, totalizando 120 amostras.

```
import math

sinSample = []

for i in range(0,361,3):
    print(i)
    print(math.sin(math.radians(i)))
```

```
sinSample.append(round(math.sin(math.radians(i))*409.5, 3))
print(len(sinSample))
print(sinSample)
```

Com as amostras geradas o programa arduino calcula o período de amostragem baseado na quantidade de amostras a partir da equação abaixo.

$$\text{período de amostragem} = \frac{\text{período seno}}{\text{nro amostras}} \times 1000000 \quad (3.1)$$

O período de amostragem está sendo multiplicado por um milhão para que ele esteja na casa dos microsegundos. Com o período de amostragem definido o programa entra em um *loop* infinito que verifica por meio da função *micros()*, que retorna a quantidade de microsegundos desde que o programa foi iniciado, se *micros()* menos o momento da última amostragem é maior ou igual ao período de amostragem. Quando essa condição é satisfeita o momento da última amostragem é atualizado com o valor de *micros()* e, por intermédio de comunicação *I2C* o Arduino escreve o valor da amostra multiplicado pela tensão escolhida no registrador do *DAC*.

Caso o usuário queira modificar o valor da amplitude ou da frequência do sinal ele deve fazer uma transição de tensão alta para baixa nos pino digital 3 do Arduino. Isto irá gerar uma interrupção e durante a rotina de tratamento desta o novo valor da amplitude ou da frequência é setado. A interrupção é gerada manualmente, o usuário deverá selecionar os valores em dois potenciômetros (um para a amplitude e um para a frequência) e depois esse deve clicar em uma chave táctil que irá gerar a interrupção. Será realizada uma conversão dos sinais analógicos dos potenciômetros para sinais digitais (respeitando as escalas de 0-5V para amplitude e 0-50Hz para a frequência) e, então, os valores destes parâmetros serão atualizados.

Foi utilizada comunicação serial para que o módulo controlador externo modifique os parâmetros de funcionamento do gerador de sinais senoidais. Em cada iteração do *loop* do *software* do arduino é verificado o número de *bytes* disponíveis para leitura da porta serial, se este número for maior ou igual a 3 condições são aplicadas ao valor lido para definir a operação a ser realizada pelo gerador de sinais: modificar amplitude (caso o valor esteja entre 200 e 206, é subtraído 200 do valor e o resto é o novo valor de amplitude), frequência (caso o valor esteja entre 100 e 151, é subtraído 100 do valor e o resto é o novo valor de frequência), ligar (caso o valor seja igual a 300) ou desligar (caso o valor seja igual a 400).

3.2.3 O circuito somador e o amplificador inversor

Neste ponto o sinal de saída do *DAC* é uma senoide que apresenta 2,5V de *offset* com amplitude igual à metade da amplitude selecionada pelo usuário. Para corrigir isso foi usado o circuito integrado TL082 que faz o uso de dois amplificadores operacionais, no primeiro op-amp foi montado um amplificador somador para somar -2,5V a este sinal, assim o transformando em um sinal *zero-mean*. Para que o sinal apresente a amplitude desejada só falta amplificar o sinal, isto é feito por intermédio de um amplificador inversor com ganho 2. O esquemático do circuito pode ser visto na Figura 3.3.

Este circuito, assim como o das fontes, foi soldado em uma placa de circuito impresso e fixado a uma outra base de eucatex e acrílico junto ao Arduino Pro Mini responsável por gerar o sinal senoidal deste módulo. A Figura 3.4 mostra este sistema.

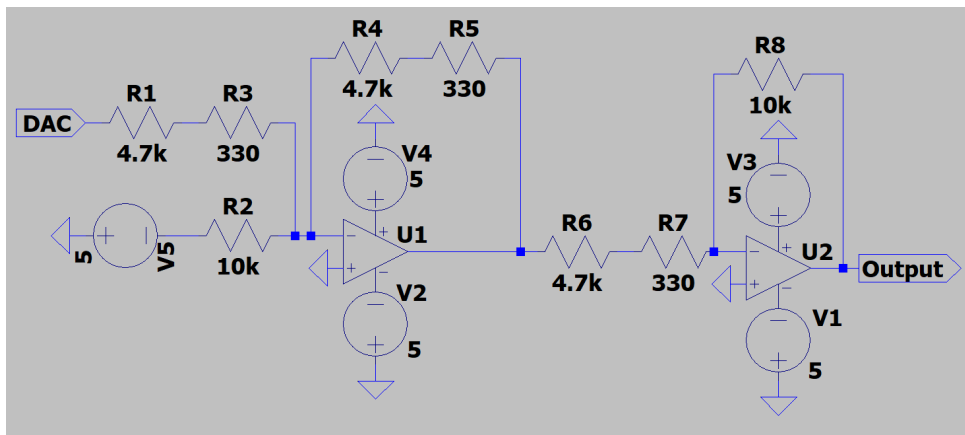


Figura 3.3: Esquemático do circuito somador e do amplificador inversor.

3.3 Sensor de fluxo

Foi desenvolvido um *software* para um *Arduino Pro Mini* com o objetivo de realizar a leitura do sensor de fluxo (SFM3000). A comunicação entre o mestre (Arduino) e escravo (SFM3000) ocorreu por meio de protocolo I2C. Primeiramente, na fase de *setup* do programa, a comunicação com o escravo é estabelecida pelo endereço presente no *datasheet* do sensor, 0X40, após isso o mestre envia o comando 0x1000 que inicia a medição contínua por parte do sensor.

Na função de *loop*, para cada iteração deste, primeiramente é realizada uma requisição de dois *bytes* do endereço 0x40 (endereço do sensor de fluxo), no primeiro *byte* recebido é realizado um *shift* de oito *bits* para a esquerda e este valor é somado ao segundo *byte* recebido do escravo. Este valor é aplicado no parâmetro “valor medido” da Equação 2.1

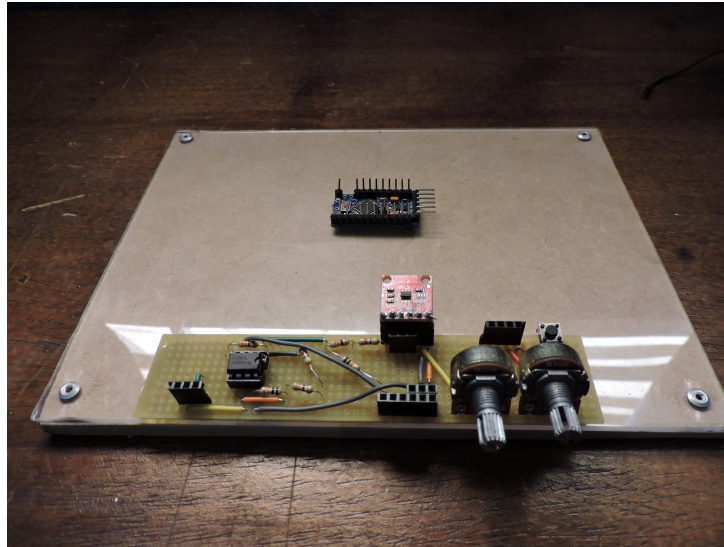


Figura 3.4: Gerador de sistemas senoidais montado.

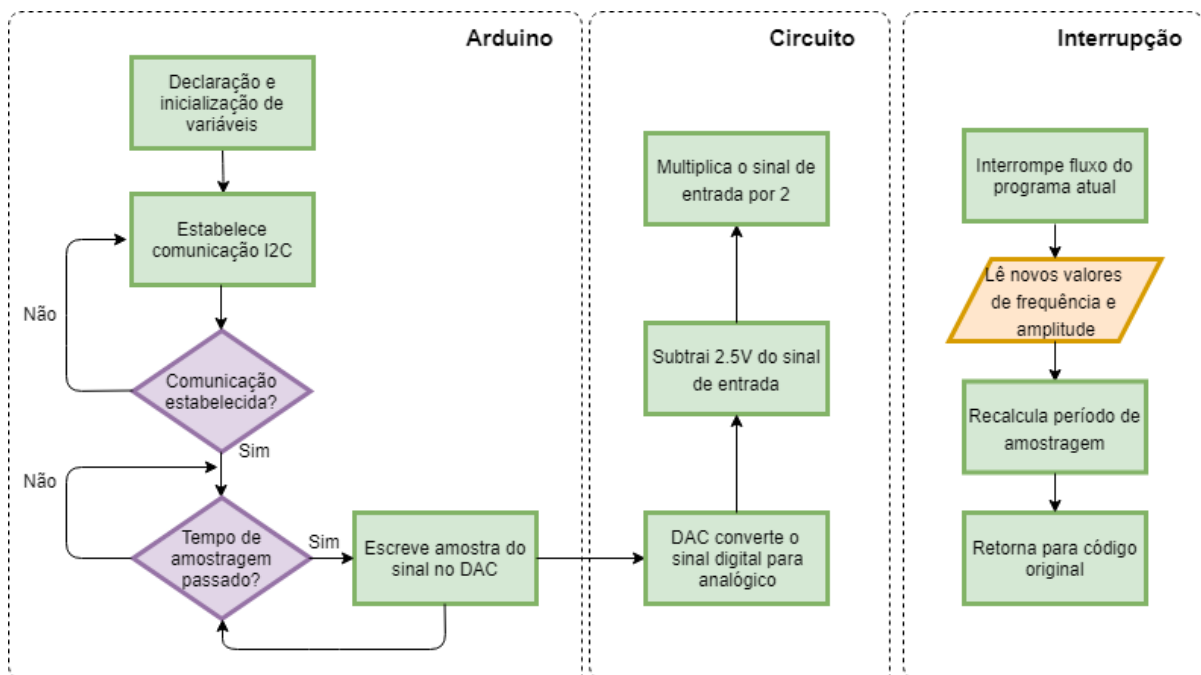


Figura 3.5: Fluxograma do gerador de sinais senoidais.

e o resultado desta equação é uma medida do fluxo de ar no tubo endotraqueal naquele instante. Cada valor aferido é então alimentado para o módulo de controle externo por meio de comunicação serial.

3.4 Sensor de pressão

Como foi discutido no capítulo 2 o sensor de pressão utilizado no projeto, MPX2010, é um sensor diferencial, que tem como saída a diferença de pressão entre os seus dois lados. Neste trabalho um dos lados fica destampado, logo fica sujeito à pressão atmosférica, e o outro é ligado ao tubo endotraqueal.

Antes de tudo foi desenvolvido um circuito responsável, inicialmente, por realizar a operação de diferença entre os valores lidos em cada um dos dois lados do sensor de pressão. Isto foi feito usando o Circuito Integrado (CI) INA118P, um amplificador de instrumentação, que recebe, como entrada, as saídas V^+ e V^- do sensor de pressão e, este CI, tem como saída a diferença entre estes valores amplificados pelo ganho do INA, dado pela Equação 3.2. O valor de R_g no circuito montado pode ser modificado por um trimpot de $10K\Omega$.

$$G = 1 + \frac{50K\Omega}{R_g} \quad (3.2)$$

Para que a simetria dos valores lidos do sensor seja mantida, foi necessário o uso da fonte simétrica de 5VDC para alimentar o circuito. Como o dispositivo responsável por realizar a leitura dos valores aferidos pelo sensor de pressão foi um *Arduino Pro Mini* e, como já foi discutido anteriormente, sua faixa de operação está entre 0 e 5V foi necessário o uso de um retificador de precisão. Este, então, foi utilizado para remover a componente negativa do sinal de saída do circuito. O retificador foi montado utilizando o amplificador operacional TL082. O esquemático do circuito pode ser visto na Figura 3.6. O circuito foi soldado em uma placa de circuito impresso e colado na mesma base de acrílico e eucatex do gerador de sinais senoidais, a placa montada é exibida na Figura 3.7.

3.4.1 A calibração

Para fornecer ao *software* os valores de leitura de saída do circuito do sensor de pressão em escala foi necessário calibrar o sistema. Foi observado que o Amplificador Operacional estava saturando em 3,67V e, optou-se que a pressão máxima a ser aferida seria de 150mmHg. Logo, a calibração foi feita de forma que quando a diferença entre a pressão aplicada à entrada positiva do sensor de pressão e à entrada exposta à pressão atmosférica fosse amplificada, o valor de saída do circuito seria de 3,5V à uma pressão de 150mmHg. Como é possível ver na Figura 3.8 isso foi feito usando um esfigmomanômetro para aferição da pressão no sistema, uma pera de pressão para aplicar pressão a esse e um multímetro para verificar a tensão de saída do circuito à medida que R_g era alterado no trimpot.

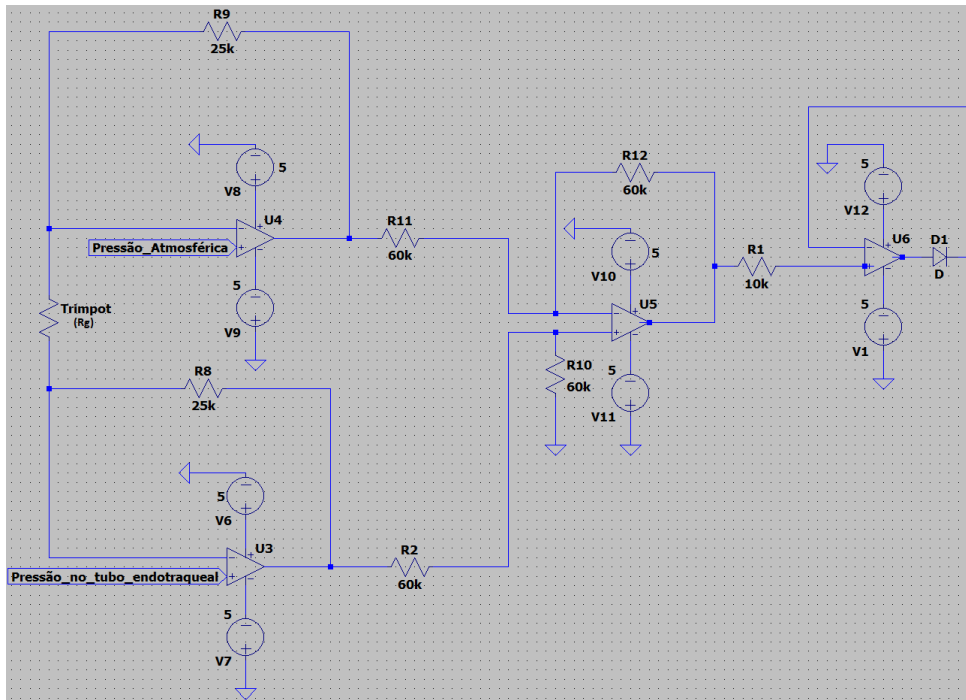


Figura 3.6: Esquemático do circuito montado para amplificar a diferença das leituras do sensor de pressão.

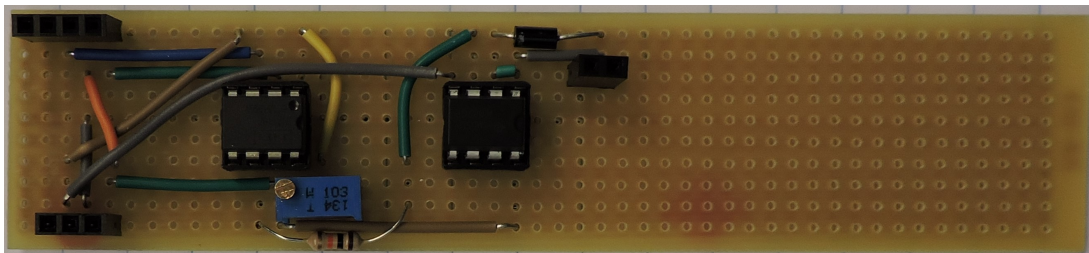


Figura 3.7: Circuito do sensor de pressão montado.

3.4.2 O software

O *software* utilizado para a leitura do circuito fez uso de um dos pinos analógicos do Arduino, estes são capazes de realizar operações de *Analog-to-Digital Converter (ADC)*, isto é, quando uma voltagem analógica é conectada ao pino analógico do Arduino e é realizada a operação de conversão, a saída é um valor digital diretamente proporcional à tensão. Como o Arduino possui um ADC de 10 *bits*, esse é capaz de detectar 2^{10} (1024) níveis analógicos discretos.

Como o *software* deve ler valores de 0V (0mmHg) a 3,5V (150mmHg) foi realizada uma simples regra de três para obter o valor digital máximo de leitura por parte do Arduino ($3,5 * 1024 = 5X$). Este valor foi “716,8”, também foi necessário converter mmHg para

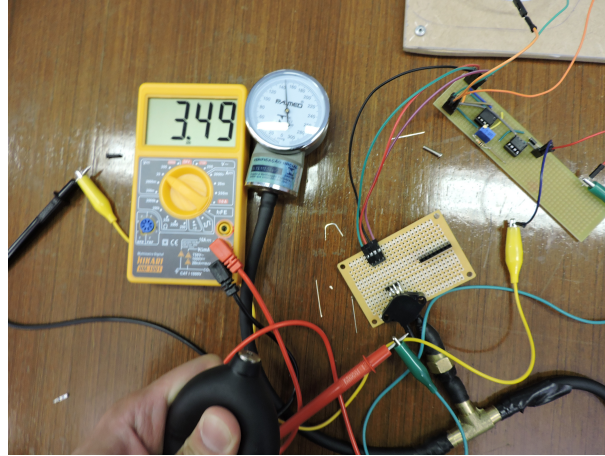


Figura 3.8: Calibração do circuito do sensor de pressão.

cmH₂O, unidade de medida de pressão escolhida, logo, “150mmHg = 203,926cmH₂O”. Com estes dois valores a leitura analógica foi convertida para cmH₂O pela Equação 3.3. Para que uma unidade controladora receba estas leituras, essa (mestre) deve estabelecer comunicação *I2C* com o Arduino (escravo) e realizar a requisição das leituras.

$$leitura\ em\ cmH_2O = \frac{leitura\ analógica}{716,8} \times 203,926 \quad (3.3)$$

Foi também utilizado um módulo *buzzer*. Este é controlado por uma porta digital configurada como output, de forma que esta ativa o *buzzer* caso a pressão aferida pelo sensor ultrapasse um valor definido no *software*, 50cmH₂O, servindo assim como um alerta caso o sistema atinja níveis perigosos de pressão.

3.5 A montagem do ventilador

Com todas as partes e componentes prontos o processo de montagem foi realizado seguindo o esquemático da Figura 3.9.

Todas as partes que compõe o gerador de funções foram conectadas e alimentadas pela fonte simétrica de 5V. Para a conexão entre o gerador e o amplificador foi usado um cabo RCA, este teve um dos conectores removidos e uma parte dos fios de ground e de sinal desemcapados nos quais jumpers foram soldados e isolados com fita isolante. Os jumpers foram conectados ao gerador de sinais e o conector do cabo RCA foi conectado ao amplificador.

O amplificador foi alimentado por uma fonte simétrica de 12V e foi conectado ao alto-falante. O tubo endotraqueal é conectado ao diafragma removível, preso à caixa de sustentação na qual fica o alto-falante. São conectados, no início deste tubo, uma

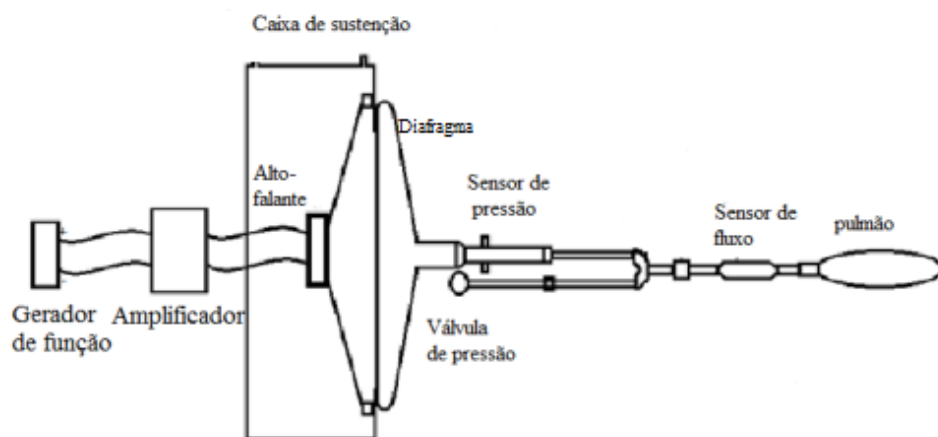


Figura 3.9: Esquemático do projeto (Fonte: [30]).

mangueira ligada ao compressor de ar e um conector tipo T, no qual, em um dos outros lados, foi plugado o esfigmomanômetro e no outro o sensor de pressão já ligado ao circuito e ao arduino para receber as leituras.

O sensor de fluxo, também conectado a um Arduino, foi preso, em um lado, com uma braçadeira ao tubo endotraqueal e no outro lado ao pulmão artificial. A válvula de pressão foi mantida meio aberta para controle de fluxo e de pressão.

Com todo o sistema montado e funcional, Figura 3.10, o próximo passo foi a coleta de dados.

3.6 A coleta de dados

A proposta da coleta de dados foi sujeitar o sistema a diversos valores amplitude e de frequência pelo gerador de sinais senoidais e, então, salvar as leituras dos sensores de pressão e de fluxo. Os testes foram realizados a 5 frequências diferentes para 3 valores de amplitude, os valores escolhidos foram: frequência[1Hz, 5Hz, 9Hz, 12Hz e 15hz] e amplitude[1V, 3V e 5V].

Ao final de cada teste (e.g. 9Hz à 3V) de dez segundos de duração, os valores dos sensores de pressão e de fluxo foram salvos em arquivos separados, de forma que, ao final dos testes, 30 arquivos foram obtidos.

Para facilitar a análise destes dados foram geradas 15 figuras, totalizando 30 gráficos, usando o *script* em python abaixo. Este *script* percorre todos os arquivos gerados agrupando, em uma figura, os gráficos dos sensores de fluxo e de pressão de cada teste (e.g. 1Hz e 5V). Foi usada uma quantidade reduzida de amostras em cada gráfico para facilitar a visualização dos formatos de onda.

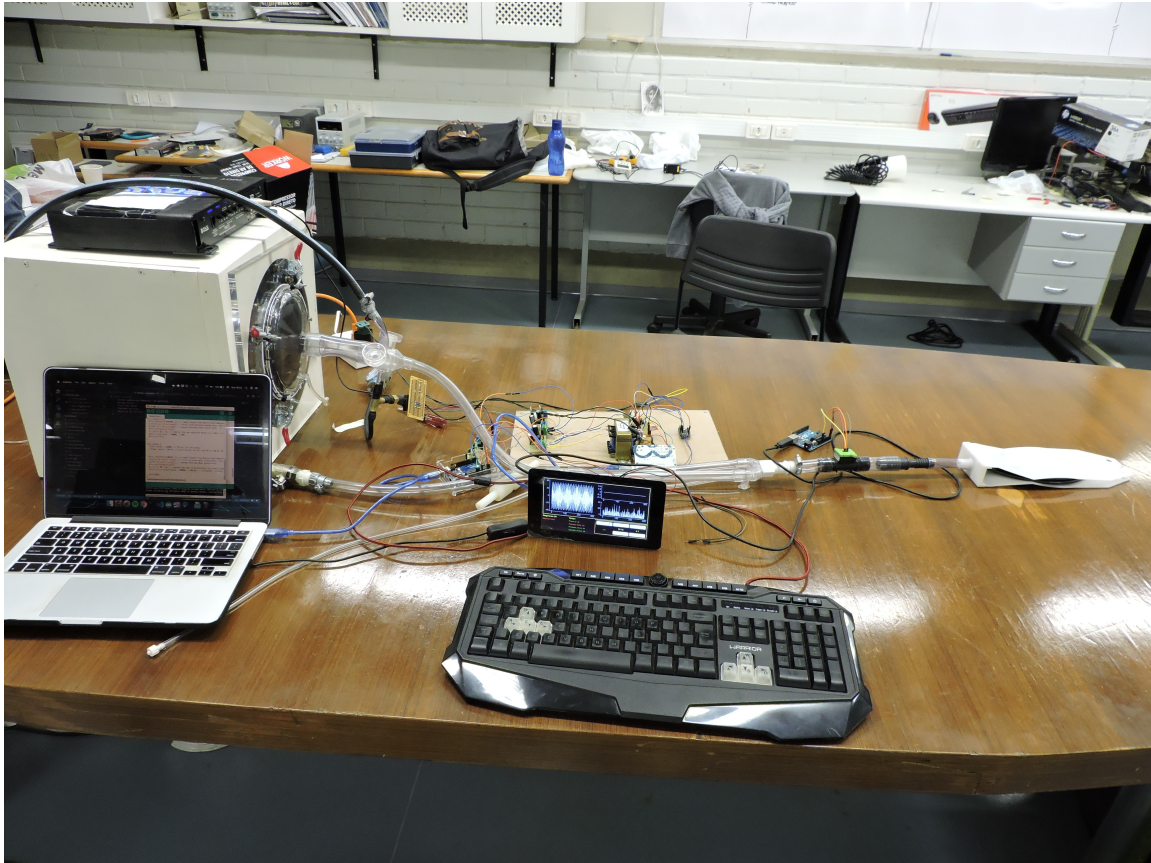


Figura 3.10: Projeto montado.

```
import matplotlib.pyplot as plt
import numpy as np

freqValues = [1, 5, 9, 12, 15]
ampValues = [1, 3, 5]
valsToPlot = []
valsLim = 0
meanValue = 0
deviation = 0
ymax, ymin = 0, 0

for m in freqValues:
    for n in ampValues:
        fileName = ['pressao' + str(n) + 'amp' + str(m) + 'hz',
                    'fluxo_' + str(n) + 'amp_' + str(m) + 'hz']
```

```

titles = ['Fluxo(SLM)', 'Presso(cmH\u20820)']
for x in fileName:
    for line in open(x + '.txt', 'r'):
        values = [float(s) for s in line.split("\r\n")]
        valsToPlot.append(values)
        valsLim = valsLim + 1
        if 'fluxo' not in x:
            if valsLim == 502:
                break
        else:
            if valsLim == 200:
                break
ymax = max(valsToPlot)
ymin = min(valsToPlot)
meanValue = round(np.mean(valsToPlot), 2)
deviation = round(np.std(valsToPlot), 2)
if 'fluxo' in x:
    plt.subplot(2, 1, 1)
    plt.plot(valsToPlot, color = '#a08dc9', linestyle = 'solid', linewidth
            = 1.0)
    plt.title(titles[0])
else:
    plt.subplot(2, 1, 2)
    plt.plot(valsToPlot, color = '#96c685', linestyle = 'solid', linewidth
            = 1.0)
    plt.title(titles[1])
plt.annotate('Mdia: ' + str(meanValue) + ', Desvio padro: ' +
            str(deviation) +
            ', ymax: ' + str(ymax) + ', ymin: ' + str(ymin), xy=(1, 0),
            xycoords='axes fraction',
            fontsize=12, horizontalalignment='right', verticalalignment='bottom')
if 'fluxo' not in x:
    plt.xlim((0,501))
else:
    plt.xlim((0,199))
plt.grid(True)
valsToPlot = []
valsLim = 0
plt.tight_layout()

```

```
plt.savefig(str(n) + 'amp_' + str(m) + 'freq.png',  
           bbovalsToPlot_inches='tight')  
plt.clf()
```

Capítulo 4

Resultados

Neste capítulo são discutidos os resultados obtidos no final do capítulo 3.

4.1 Análise dos gráficos

Na Figura 4.1 observa-se um bom comportamento para as leituras de fluxo, visto que a fase inspiratória e expiratória foram constantes entre 10 e -10SLM. A pressão apresentou uma *Mean Airway Pressure (MAP)* de 33.22cmH₂O e uma amplitude (ΔP), baseada nos valores ymax e ymin, de 27,88cmH₂O, a pressão não saiu como esperada, visto que o formato de onda desejado seria similar a senoide que controla o movimento do alto-falante.

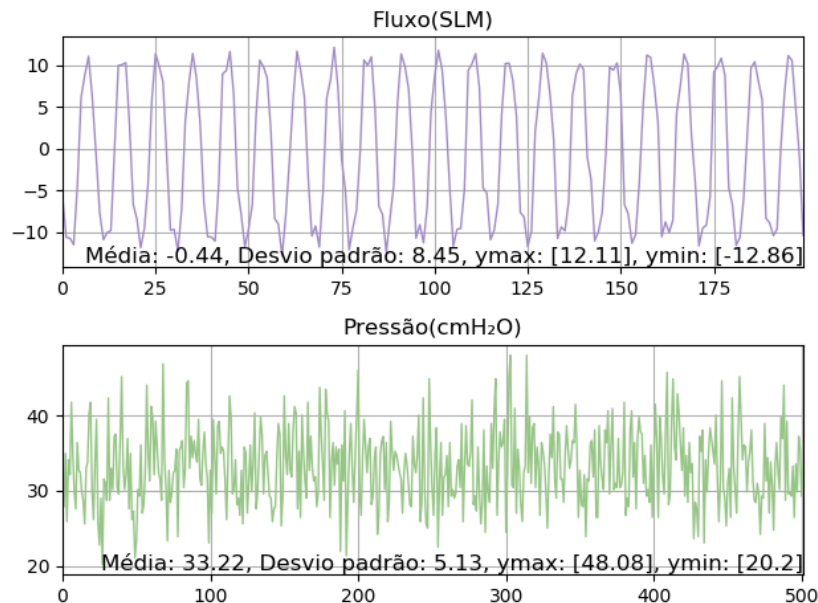


Figura 4.1: Gráfico dos dados dos sensores de fluxo e de pressão à 5Hz e 1V de amplitude.

Como é possível ver na Figura 4.2 o aumento de 4Hz na entrada do sistema provocou um aumento considerável no fluxo e a pressão manteve um comportamento não ideal.

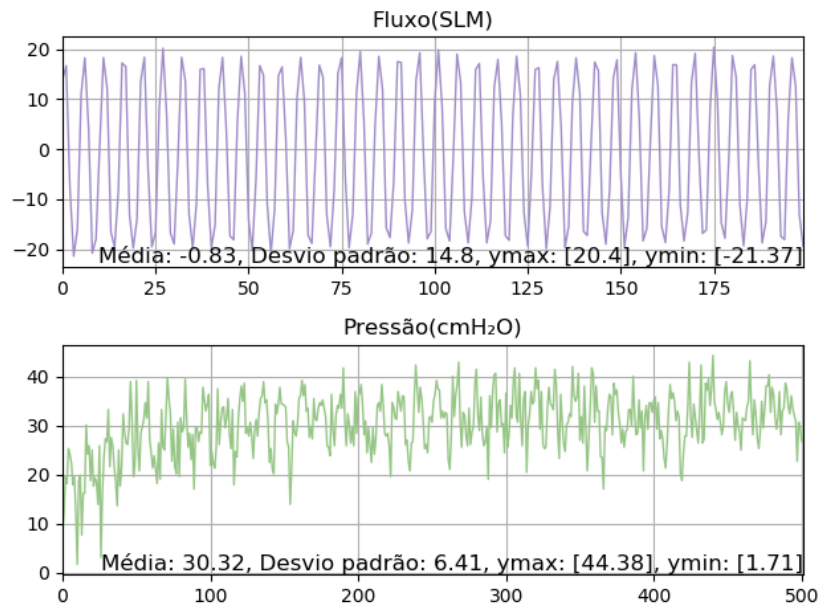


Figura 4.2: Gráfico dos dados dos sensores de fluxo e de pressão à 9Hz e 1V de amplitude.

À 12Hz o fluxo aumentou ainda mais e a pressão manteve uma MAP próxima a 33cmH₂O.

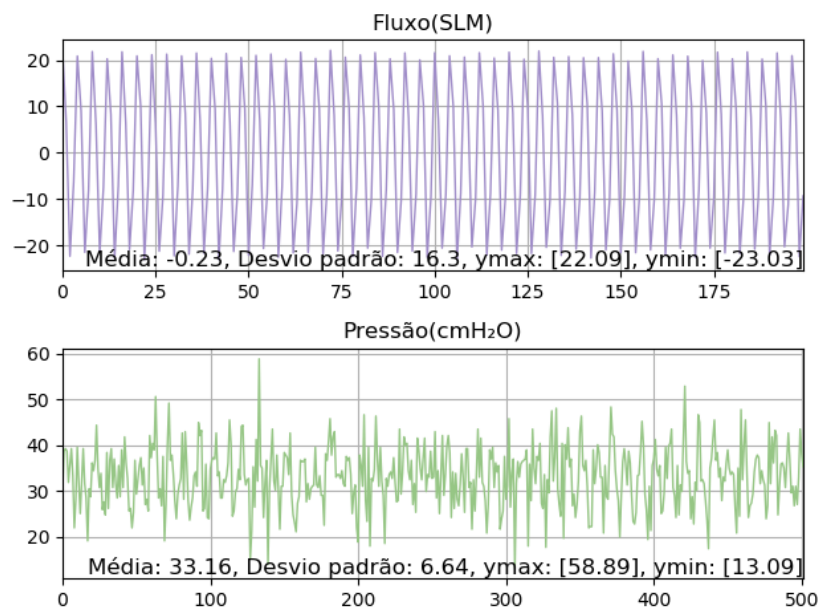


Figura 4.3: Gráfico dos dados dos sensores de fluxo e de pressão à 12Hz e 1V de amplitude.

Com o sistema sendo alimentado por uma senoide de 1V de amplitude e 15Hz o fluxo apresentou um desvio padrão mais de duas vezes maior que o observado na primeira aferição e passou a sofrer variações indesejadas. A pressão se manteve praticamente constante ao longo de todas as medidas a 1V de amplitude.

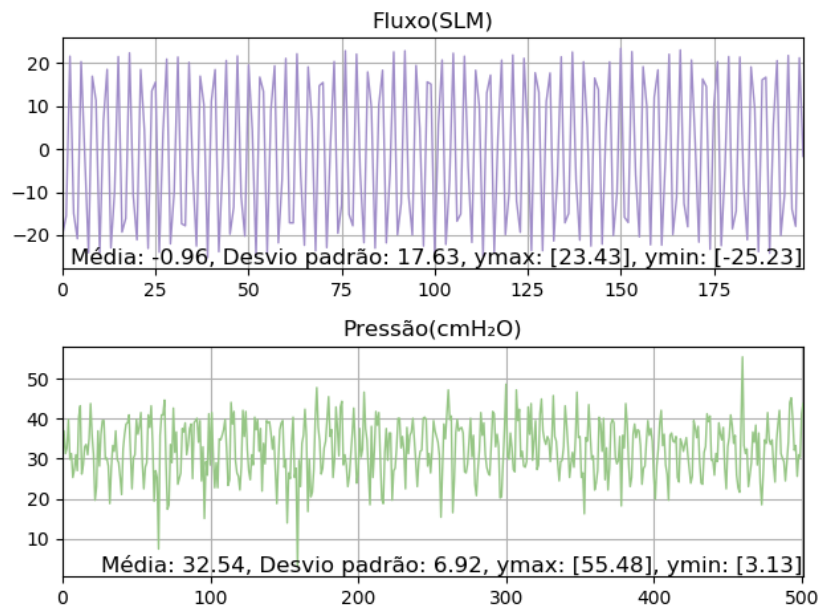


Figura 4.4: Gráfico dos dados dos sensores de fluxo e de pressão à 15Hz e 1V de amplitude.

Com o aumento da amplitude, fica claro, observando a Figura 4.5, uma melhora na onda aferida para a pressão. Esta apresentou um valor médio de 33,06cmH₂O e o fluxo teve um aumento bastante significativo em relação ao medido a 1V de amplitude e 5Hz, isso acontece porque o aumento da amplitude faz com que o pistão responsável pelas oscilações se desloque mais.

Na Figura 4.8, a pressão apresenta um comportamento esperado, visto que o sinal gerado é, até agora, o mais próximo de uma senoide. A amplitude do sinal de pressão foi de 54,34cmH₂O. O fluxo, assim como a aferição realizada com 1 Hz e 1V, não foi estável.

Comparando o sinal de pressão em 5Hz a 1, 3 e 5V é possível observar um aumento na amplitude e da qualidade do sinal à medida que a amplitude aumenta. No fluxo o sinal se manteve estável em todas as amplitudes testadas para 5Hz.

Os melhores sinais foram obtidos quando as ondas senoidais foram aplicadas à 12Hz. O fluxo se manteve constante e o sinal da pressão melhorou à medida que a amplitude de movimento do pistão aumentou. Amplitudes mais altas levam a maior retirada de CO₂ devido ao aumento do volume corrente. O fluxo, como esperado, aumentou significativamente da leitura com 1V para a com 5V, na primeira o fluxo variou entre 22,09SLM e -23,03SLM e na última leitura a 12Hz o fluxo variou entre 40,46SLM e -44,57SLM.

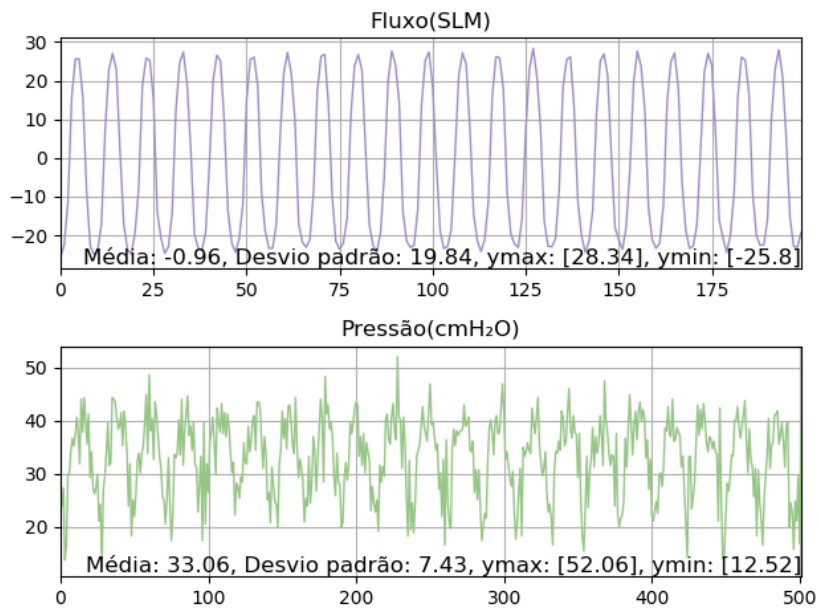


Figura 4.5: Gráfico dos dados dos sensores de fluxo e de pressão à 5Hz e 3V de amplitude.

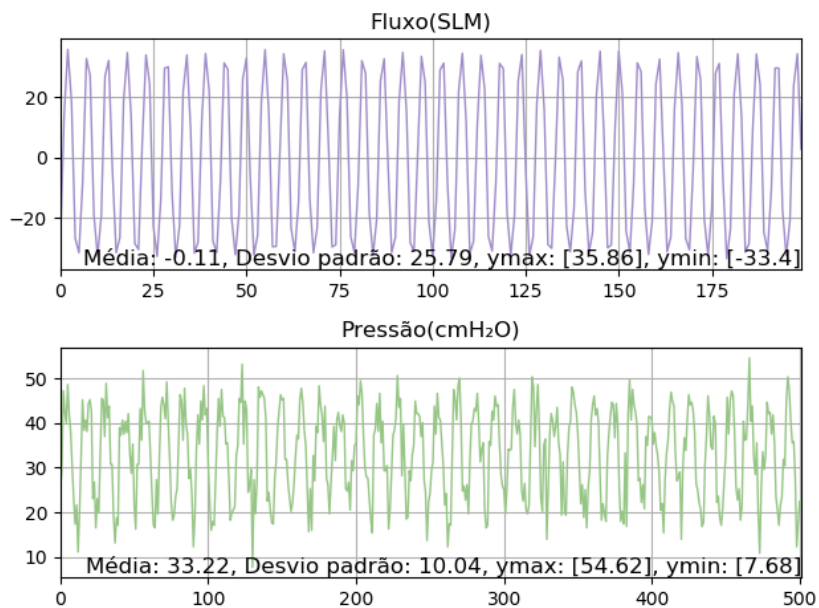


Figura 4.6: Gráfico dos dados dos sensores de fluxo e de pressão à 9Hz e 3V de amplitude.

Na Figura 4.12, assim como nas outras leituras a 15Hz o fluxo e a pressão apresentaram redução da amplitude dos sinais de forma periódica. Em todos os testes foi possível observar que o aumento da frequência e da amplitude do sinal que controla a oscilação aumentaram o Δp .

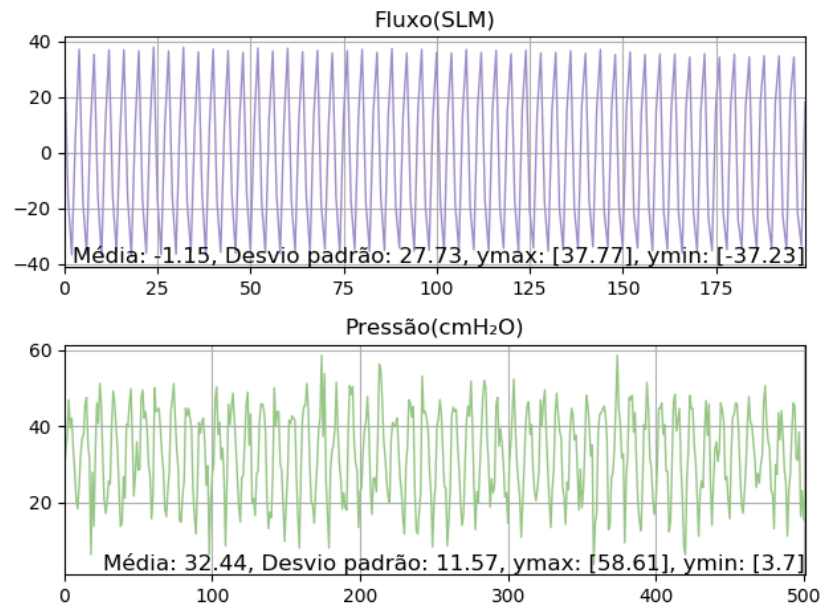


Figura 4.7: Gráfico dos dados dos sensores de fluxo e de pressão à 12Hz e 3V de amplitude.

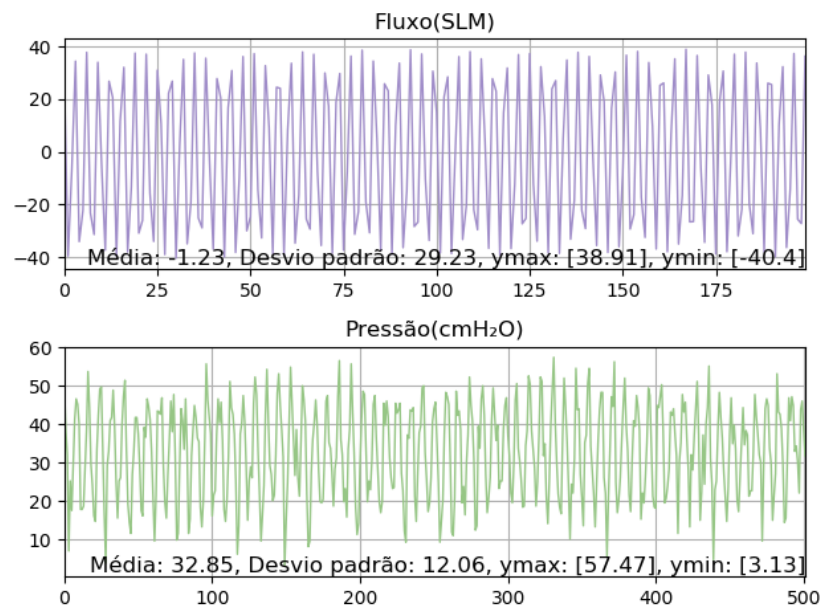


Figura 4.8: Gráfico dos dados dos sensores de fluxo e de pressão à 15Hz e 3V de amplitude.

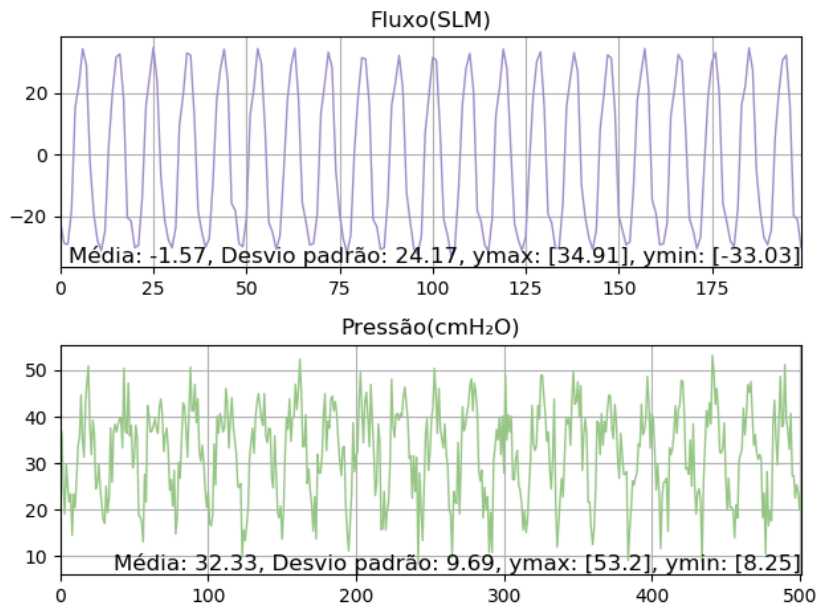


Figura 4.9: Gráfico dos dados dos sensores de fluxo e de pressão à 5Hz e 5V de amplitude.

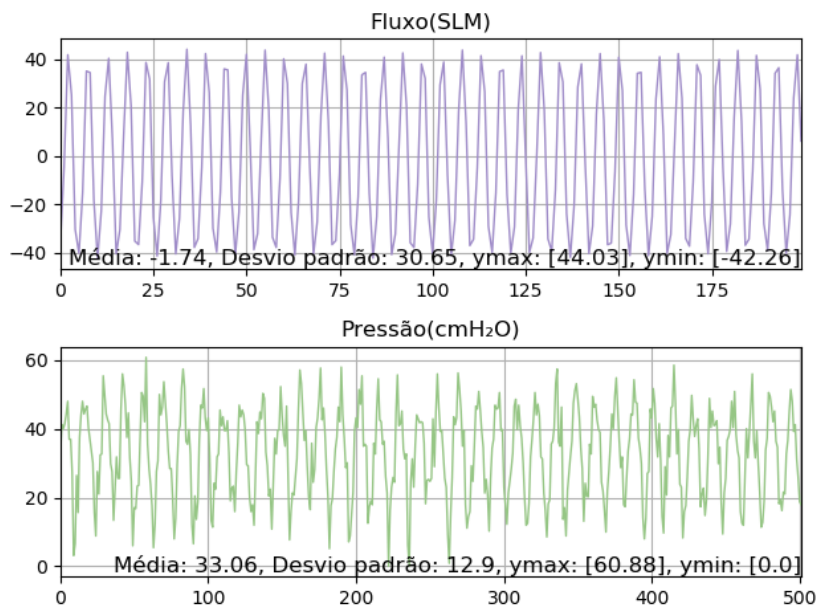


Figura 4.10: Gráfico dos dados dos sensores de fluxo e de pressão à 9Hz e 5V de amplitude.

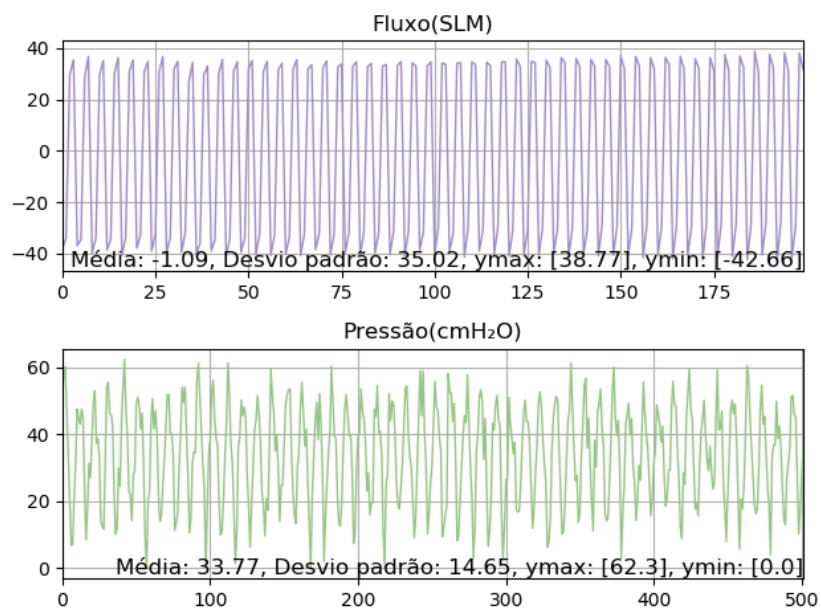


Figura 4.11: Gráfico dos dados dos sensores de fluxo e de pressão à 12Hz e 5V de amplitude.

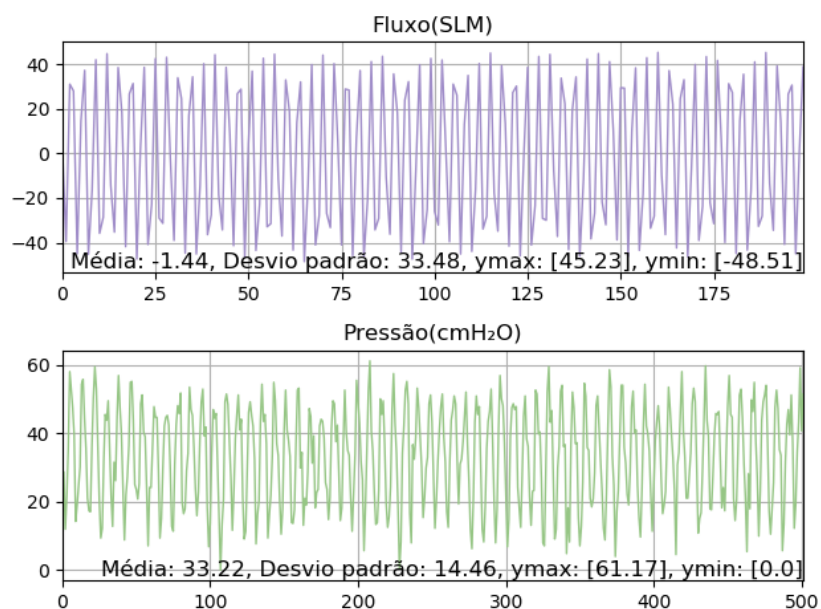


Figura 4.12: Gráfico dos dados dos sensores de fluxo e de pressão à 15Hz e 5V de amplitude.

Capítulo 5

Conclusão

O ventilador pulmonar oscilatório de alta frequência é um aparelho responsável por realizar tanto as fases inspiratórias quanto as expiratórias da respiração humana. Esse dispositivo é amplamente usado no tratamento de recém-nascidos prematuros com problemas respiratórios.

Com o intuito de melhorar o projeto do *HFOV* foram desenvolvidos três módulos. Um gerador de sinais senoidais com parâmetros variáveis para definir o funcionamento do oscilador e, assim, controlar o funcionamento do sistema; foi desenvolvido um script para ler os resultados fornecidos por um sensor de fluxo responsável por averiguar se o fluxo de ar pelo tubo endotraqueal funcionou como o esperado; e um módulo responsável por verificar se a pressão do sistema se mantinha em uma pressão média das vias aéreas desejada com as variações de pressão definidas pelo comportamento da senoide do primeiro módulo. Todos estes módulos funcionam em tempo real e são capazes de se comunicar com uma unidade controladora.

Os resultados observados foram especialmente satisfatórios quando foram aplicadas frequências de 12Hz para os três níveis de amplitude testados. As leituras dos sensores de pressão se comportaram melhor com amplitudes da senoide mais altas. Foi possível visualizar que o pulmão artificial se manteve meio cheio pela *MAP* e, observar, que o aumento da amplitude da senoide aumentou a amplitude de movimento do pulmão (Δp). As leituras também claramente exibem formatos senoidais definidos pelo oscilador.

5.1 Trabalhos futuros

Como o projeto atual ainda se encontra em uma etapa inicial várias mudanças podem ser propostas. O uso de mais um sensor de pressão no final do tubo endotraqueal seria interessante para melhor identificar vazamentos ao longo do mesmo; integrar um umidificador aquecido ao sistema para aproximar o projeto a um ventilador comercial; automatizar o

controle da válvula de pressão; desenvolver uma estrutura de suporte para o ventilador e seus componentes.

Referências

- [1] Enderle, John e Joseph Bronzino: *Introduction to biomedical engineering*. Academic press, 2012. 1
- [2] *Biomedical engineering: What is it and what are the career opportunities?* <https://www.mendeley.com/careers/article/biomedical-engineering-career-opportunities/#What%20is%20Biomedical%20Engineering>. 1
- [3] mediLexicon: *Definition: 'artificial ventilation'*. <https://www.medilexicon.com/dictionary/98071>. 4
- [4] Baker, A Barrington: *Artificial respiration, the history of an idea*. *Medical history*, 15(4):336–351, 1971. 4, 5
- [5] Geddes, LA: *The history of artificial respiration [retrospectroscope]*. *IEEE Engineering in Medicine and Biology Magazine*, 26(6):38–41, 2007. 5
- [6] Hewa: *Iron lung*, December 2011. <https://en.wikipedia.org/wiki/File:Museum-gt-eiserne-lunge.jpg>. 6
- [7] Young, JD e MK Sykes: *Assisted ventilation. 1. artificial ventilation: history, equipment and techniques*. *Thorax*, 45(10):753, 1990. 6
- [8] Media, Nucleus Medical: *Intubation and mechanical ventilation*. <https://ebSCO.smartimagebase.com/intubation-and-mechanical-ventilation/view-item?ItemID=70896>. 6
- [9] Society, American Thoracic: *Mechanical ventilation*. *American Journal of Respiratory and Critical Care Medicine*, 196:3–4, 2017. 6
- [10] *Mechanical ventilation: Conventional and non-conventional*. <https://study.com/academy/lesson/mechanical-ventilation-conventional-non-conventional.html>. 6
- [11] Beardsell, Iain: *Beardsell, I et al: MCEM Part A:MCQs*. Royal Society of Medicine Press, 2009. 6
- [12] Borrello, Michael A: *Closed loop control system for a high frequency oscillation ventilator*, janeiro 2011. US Patent 7,861,716. 7, 8

- [13] Hospitals, St George/Sutherland e Health Services: *High frequency oscillating ventilation (hfov) – icu –st george hospital*, 2013. 7
- [14] Bennarsten, Johan: *High-frequency oscillator ventilator*, julho 2003. US Patent 6,595,213. 7, 8
- [15] Boussignac, Georges: *Tracheal catheter*, junho 2003. US Patent 6,575,166. 8
- [16] Western Australia North Metropolitan Health Service, Government of: *Ventilation: High frequency oscillation ventilation*. 8
- [17] Lampland, Andrea L e Mark C Mammel: *The role of high-frequency ventilation in neonates: evidence-based recommendations*. Clinics in perinatology, 34(1):129–144, 2007. 8
- [18] Bennarsten, Johan: *High-frequency oscillation patient ventilator system*, fevereiro 2004. US Patent 6,694,978. 8
- [19] *Understanding high-frequency oscillatory ventilation*. <https://www.americannursetoday.com/understanding-high-frequency-oscillatory-ventillation/>. 8
- [20] Incorporated, Microchip Technology: *Datasheet mcp4725, 12-bit digital-to-analog converter with eeprom memory in sot-23-6*, 2009. 9
- [21] *Mcp4725*. <https://uploads.filipeflop.com/2017/07/02-19.png>. 9
- [22] SENSIRION: *Datasheet sfm3000, low pressure drop digital flow meter*, 2016. 9
- [23] Semiconductor, Freescale: *Datasheet mpx12, 10 kpa uncompensated silicon pressure sensors*, 2009. 10
- [24] Banzi, Massimo e Michael Shiloh: *Getting started with Arduino: the open source electronics prototyping platform*. Maker Media, Inc., 2014. 11
- [25] Arduino, Store Arduino: *Arduino*. Arduino LLC, 2015. 11
- [26] *Arduino pro mini*. https://uploads.filipeflop.com/2017/07/1AC09_Arduino_Pro_Mini_1-1.jpg. 11
- [27] *attachinterrupt()*. <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>. 12
- [28] Kaith, Deepa, Janakkumar B Patel e Neeraj Gupta: *An introduction to functional verification of i2c protocol using uvm*. International Journal of Computer Applications, 121(13), 2015. 12, 13
- [29] *Fonte simétrica de 5v com modificações*. <http://www.electronics-lab.com/project/5v-symmetrical-regulated-power-supply/>. 15
- [30] FERREIRA, Jefferson, 2017. 23