
BOMBA PERISTÁLTICA

Relatório Final

F 530 – Instrumentação I

Renata Possobon RA 147804
re.possobon @ gmail.com



Orientador:

Prof. Dr. Varlei Rodrigues
varlei @ ifi.unicamp.br
DFA - GFNMN

Co-orientador:

Prof. Dr. Antonio Riul Junior
riul @ ifi.unicamp.br
DFA - GFNMN

Coordenador:

Prof. Dr. José J. Lunazzi
lunazzi @ ifi.unicamp.br
DFMC

1. INTRODUÇÃO

Em alguns estudos realizados pelo Grupo de Física de Nanossistemas e Materiais Nanoestruturados (GFNMN) do Departamento de Física Aplicada (DFA) é necessário que certos líquidos fluam continuamente na superfície de materiais que se encontram no interior de microcanais por algumas horas. Para isso é necessário o uso de algum tipo de dispositivo que possibilite o movimento cíclico do fluido com velocidade e vazão controladas, no nosso caso, uma bomba peristáltica. Este projeto consiste em construir uma bomba peristáltica para estes estudos utilizando a tecnologia de impressão 3D. O desenvolvimento proposto apresenta vantagens do ponto de vista do controle integrado da bomba a outros instrumentos, e também do ponto de vista econômico pelo fato da impressão 3D viabilizar o sistema com custo pelo menos cinco vezes mais barato que sistemas comerciais.

2. DESCRIÇÃO

Uma bomba peristáltica se baseia na alternância de compressões e relaxamentos de uma mangueira, permitindo a movimentação de dosagens precisas de um fluido ^[1]. O aparelho consiste basicamente de um cabeçote em forma de ferradura, um tipo de rotor com roletes e uma mangueira, que é posicionada entre o rotor e o cabeçote. A Figura 1 ilustra um esquema simplificado da bomba peristáltica. Os roletes do rotor movem-se pela mangueira, comprimindo-a e movimentando a parte comprimida. Na porção logo atrás do rolete forma-se vácuo que empurra o fluido sem que haja deslizamentos.

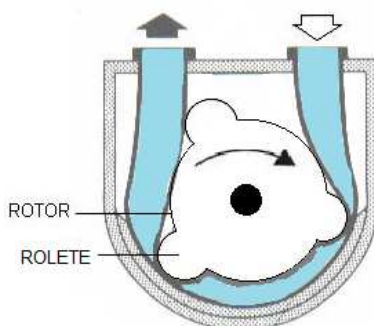


Figura 1 – Esquema simplificado de uma bomba peristáltica.

3. DESENVOLVIMENTO DO PROJETO

3.1 Introdução à impressão 3D

Para iniciar o projeto era necessário familiarizar-se com os softwares *Slic3r* ^[2] e *Pronterfacer* ^[3], que são as ferramentas básicas necessárias para a impressão 3D e também aprender os procedimentos de calibração para impressão das peças utilizadas no projeto. O *Slic3r* é o software utilizado para converter um modelo 3D em instruções para impressora. Basicamente, o *Slic3r* divide o modelo em camadas e gera percursos de preenchimento de acordo com as especificações do usuário em relação à quantidade de material depositado e velocidade de extrusão. Depois de carregar o modelo no software e configurar os parâmetros

desejados para a impressão, cria-se um código denominado *G-Code* [4] que contém todas as instruções para a impressora. Esse código é então carregado no *Pronterfacar*, software que permite comunicação com a impressora e acompanhamento do progresso da impressão, possibilitando enviar informações à impressora e gerenciar comandos.

O primeiro modelo 3D utilizado, um cubo, foi impresso a fim de permitir a familiarização com estes softwares e com o procedimento de calibração da impressora, além de permitir a busca da configuração mais apropriada para o tipo de material utilizado e para as necessidades de cada peça. Depois destes modelos, ainda foram impressos vários outros, aleatórios, até que os procedimentos para impressão fossem completamente dominados.

3.2 Peças

Após este primeiro contato com a impressão 3D, iniciou-se o trabalho com as peças necessárias para a montagem da bomba peristáltica. Tais peças tratam-se de um projeto open hardware e encontram-se disponíveis no *thingiverse* [5][6]. Os modelos das peças estão ilustrados na Figura 2.

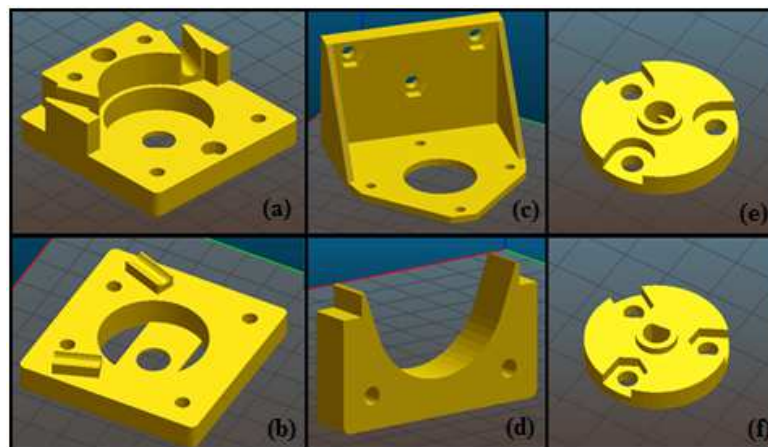


Figura 2 – Modelos das peças da bomba peristáltica. (a) Base da bomba. (b) Parte superior da bomba. (c) Suporte do motor (d) Placa de pressão da mangueira. (e) Rotor inferior. (f) Rotor superior.

Algumas dificuldades foram encontradas na impressão 3D das peças da bomba, como imprevistos com a impressora e problemas em várias impressões ilustrados na Figura 3. Após uma sucessão de problemas com a primeira impressora que utilizamos, decidiu-se trocar de equipamento (nova impressora) e usar o polímero ABS na impressão, obtendo-se o resultado ilustrado na Figura 4.



Figura 3 – Exemplos de peças que apresentaram problemas na impressão.



Figura 4 – Peças da bomba peristáltica impressas em ABS puro.

3.3 Mecânica

Após a impressão das peças a parte mecânica da bomba foi finalizada, utilizando além das peças impressas, três rolamentos (13x4x5), algumas porcas, parafusos e arruelas, um motor de passo e um tubo de silicone de 4 mm. Inicialmente, os rolamentos foram fixados entre os rotores superior e inferior, como ilustrado na Figura 5. Separadamente, o motor e a base da bomba foram fixados no suporte do motor e o conjunto formado pelos rotores e rolamentos foi encaixado no eixo do motor. A placa de pressão da mangueira foi encaixada na base e a mangueira foi posicionada de maneira adequada entre o rotor e a placa de pressão, resultando no sistema ilustrado na Figura 6 (a). Por fim, foi fixada neste sistema a parte superior da bomba, finalizando a montagem da parte mecânica, obtendo o resultado ilustrado na Figura 6 (b).



Figura 5 – Conjunto formado pelas peças rotor superior e rotor inferior com o s rolamentos. Esta peça promoverá a compressão da mangueira quando a bomba peristáltica for ligada.

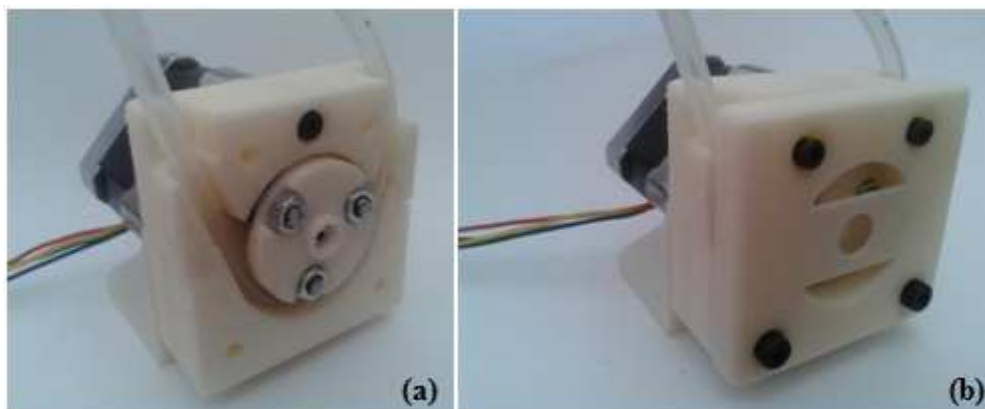


Figura 6 – Resultado obtido após a finalização da montagem da parte mecânica da bomba peristáltica. (a) Nesta imagem é possível observar a parte interna da bomba e como a mangueira é posicionada na cabeça da bomba. O conjunto de peças do rotor mais os rolamentos espremem a mangueira quando o motor é ligado, empurrando o fluído. (b) Bomba peristáltica.

3.4 Eletrônica

Ao mesmo tempo em que a parte mecânica da bomba era desenvolvida, estudou-se em paralelo o sistema Arduino [7], que é uma plataforma de desenvolvimento de código aberto com base em uma placa microcontroladora e em um ambiente de desenvolvimento que permite a implementação do código. Utilizou-se o livro “Primeiros Passos com o Arduino”, de Massimo Banzi [8], e inúmeras programações simples utilizando circuitos com diferentes componentes, como LEDs, botões de pressão e sensores de luz, por exemplo. Posteriormente, utilizando os conhecimentos adquiridos nesta fase inicial, desenvolveu-se uma eletrônica, ilustrada na Figura 7, que permite o controle da bomba baseado no Arduino e utilizando o *driver* A4988 [9][10].

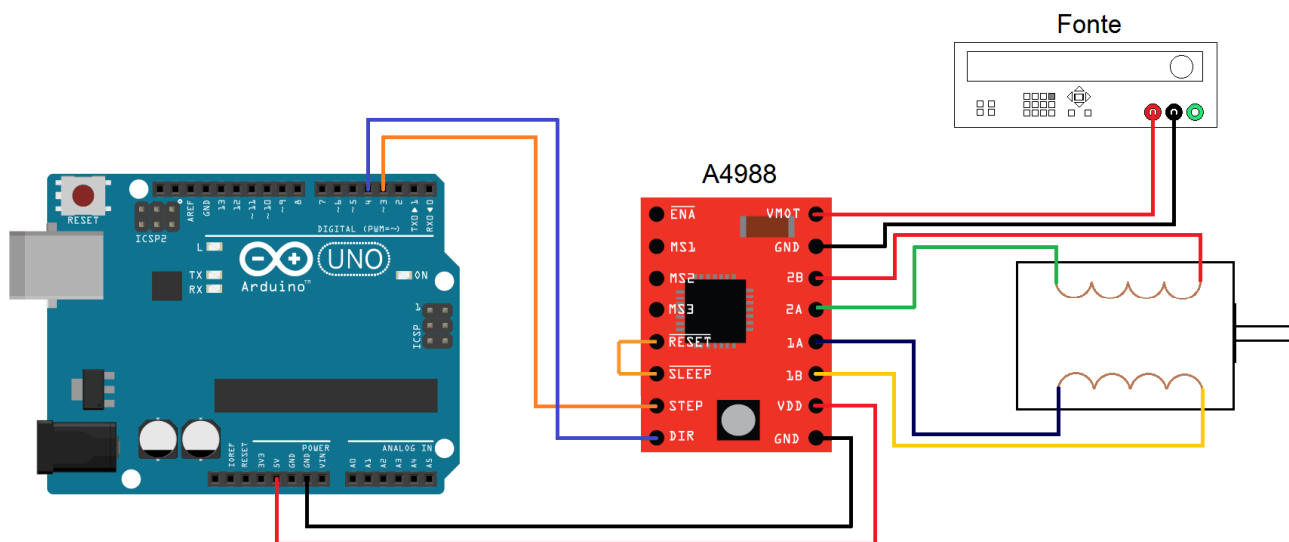


Figura 7 – Eletrônica elaborada para o controle da bomba peristáltica.

O A4988, ilustrado da Figura 8, é um *driver* que permite o controle de pequenos passos (*microstepping*) e de direção do motor de passo bipolar, utilizando uma interface simples.

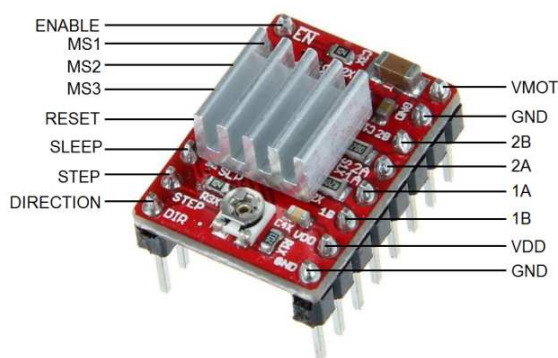


Figura 8 – Driver A4988.

Este dispositivo requer uma tensão de alimentação lógica, conectada através dos pinos VDD e GND, que pode variar entre 3 V e 5,5 V e uma tensão de alimentação do motor, conectada através de VMOT e GND, que pode variar entre 8 V e 35 V, com capacidade de condução de saída de 2 A. No caso da Figura 7, a fonte de alimentação lógica utilizada foi o próprio Arduino, que alimenta o *driver* com uma tensão de 5 V

e a alimentação do motor foi realizada por uma fonte externa, com saída de 15 V e 0,4 A. As entradas 1A e 1B do *driver* alimentam uma bobina do motor enquanto as entradas 2A e 2B alimentam a outra bobina do motor.

As entradas RESET, SLEEP e ENABLE são utilizadas para controlar os diferentes estados de energia do *driver*: RESET define o *driver* para um estado pré-definido, SLEEP coloca a placa no “modo de espera” para minimizar o consumo de energia quando o motor não está em uso e o pino ENABLE é utilizado para ligar ou desligar as saídas FET. O *jumper* utilizado entre as entradas SLEEP e RESET leva a entrada RESET para um estado ligado (*High*) e habilita a placa. O pino ENABLE não foi utilizado.

A entrada STEP recebe os pulsos que corresponderão aos passos do motor, enquanto a entrada DIRECTION determina o sentido de rotação do motor (“*High*” para sentido anti-horário e “*Low*” para sentido horário). A definição desses parâmetros é feita no Arduino. A Figura 9 ilustra o programa que foi carregado no Arduino, que define o sentido de rotação do motor através do pino 4 como anti-horário e manda os pulsos para o STEP através do pino 3.



```
Motor_Bomba | Arduino 1.6.11
Arquivo Editar Sketch Ferramentas Ajuda
Motor_Bomba $
int x;

void setup() {

  pinMode(3,OUTPUT); //STEP
  pinMode(4,OUTPUT); //DIRECTION

}

void loop() {
  digitalWrite(4,HIGH);
  for(x = 0; x < 2000; x++) {
    digitalWrite(3,HIGH);
    delayMicroseconds(200);
    digitalWrite(3,LOW);
    delayMicroseconds(200);
  }
}

Carregado.

O sketch usa 846 bytes (2%) de espaço de armazenamento para programas. O máximo
Variáveis globais usam 11 bytes (0%) de memória dinâmica, deixando 2.037 bytes p
18 Arduino/Genuino Uno em COM3
```

Figura 9 – Programa utilizado no Arduino para movimentar a bomba peristáltica.

Os motores de passo têm tipicamente uma especificação de tamanho de passo que se aplica a passos completos. O A4988 proporciona resoluções mais altas, permitindo passos intermediários, que são conseguidas energizando as bobinas com níveis de corrente intermediários. As entradas do seletor de

resolução MS1, MS2 e MS3 permitem a seleção de cinco resoluções de passo dependendo de qual combinação das posições ligado (*High*) e desligado (*Low*) destas entradas é utilizada. A Tabela 1 ilustra as possíveis resoluções de passo que podem ser configuradas no A4988.

Tabela 1 – Possíveis resoluções de passo intermediárias que podem ser configuradas ajustando as entradas MS1, MS2 e MS3 do *driver* A4988.

MS1	MS2	MS3	Resolução do microstep
Low	Low	Low	Passo completo
High	Low	Low	$\frac{1}{2}$ passo
Low	High	Low	$\frac{1}{4}$ passo
High	High	Low	$\frac{1}{8}$ passo
High	High	High	$\frac{1}{16}$ passo

Para os fins da bomba peristáltica a resolução de $\frac{1}{4}$ passo mostrou-se mais adequada. Sendo assim, utilizou-se a entrada MS2 na posição “*High*”, ou seja, a entrada foi alimentada com a tensão de 5 V do Arduino. A montagem final da bomba peristáltica está ilustrada na Figura 10.

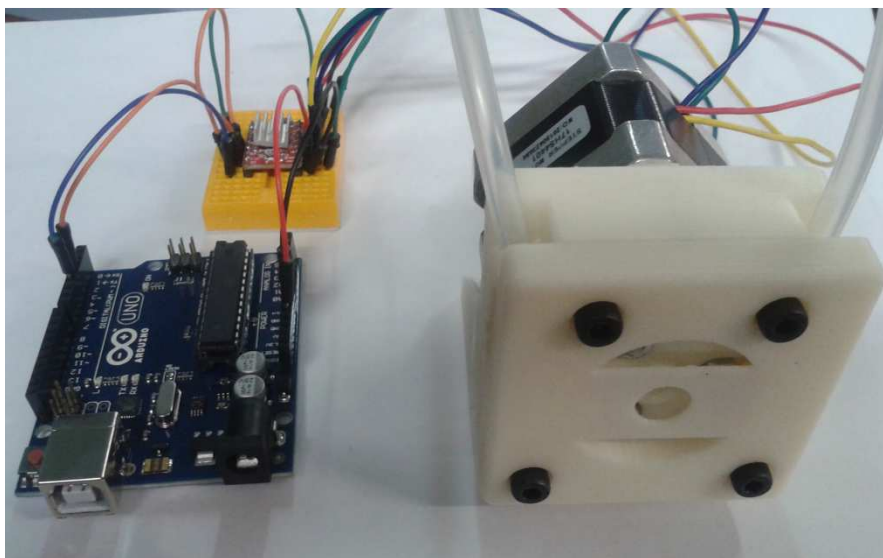


Figura 10 – Bomba peristáltica: montagem completa.

Com o programa indicado na Figura 9 carregado no Arduino e alimentando corretamente o *driver* e o Arduino, a bomba peristáltica apresentou um bom funcionamento.

4. CONSIDERAÇÕES FINAIS

O projeto foi desenvolvido dentro dos prazos e a bomba peristáltica apresentou um bom funcionamento. Esta bomba peristáltica será utilizada posteriormente para o alinhamento de nanobastões de ouro no interior de microcanais, visando amplificação de sinais Raman (SERS, do inglês *Surface Enhanced*

Raman Scattering). Este projeto está sendo desenvolvido em parceria com o Prof. Dr. Italo Mazzali (IQ, UNICAMP).

É válido ressaltar, também, que no dia 22 de novembro foi realizado o evento de Consulta à Comunidade das disciplinas orientadas pelo Prof. Dr. Lunazzi, no qual foi possível apresentar o projeto a alguns alunos e professores.

Em relação à contribuição deste projeto para minha formação, gostaria de destacar que o projeto foi de grande importância, pois proporcionou o aprendizado sobre vários equipamentos e softwares, a aplicação e aprimoramento de conhecimentos e a aquisição de novos conhecimentos e de experiência em instrumentação, além de proporcionar a interação com estudantes e especialistas de diferentes áreas de estudo que participam do grupo, que foi muito enriquecedora.

Meus orientadores concordam com o expressado neste relatório e deram a seguinte opinião: “A aluna tem comportamento exemplar, terminando o projeto com perfeição. Destacamos sua elevada capacidade para estudos avançados, disciplina, boa vontade, mantendo ótimo ambiente de trabalho com colegas no grupo de pesquisa, além de compromisso e empenho nas atividades que realiza.”

REFERÊNCIAS

- [1] SPLABOR. **Bomba peristáltica**. Disponível em: <<http://www.splabor.com.br/blog/aprendendo-mais/aprendendo-mais-o-que-e-e-como-funciona-uma-bomba-peristaltica/>>
- [2] Slic3r Manual. Disponível em: <<http://manual.slic3r.org/>>
- [3] PRINT RUN. **Pronterface**. Disponível em: <<http://www.pronterface.com/>>
- [4] RepRap. **G-Code**. Disponível em: <<http://reprap.org/wiki/G-code>>
- [5] Projeto das peças da bomba. <<http://www.thingiverse.com/thing:254956>>
- [6] Projeto das peças da bomba. <<http://www.thingiverse.com/thing:1134817>>
- [7] Arduino. Disponível em: <<https://www.arduino.cc/>>
- [8] BANZI, M. **Primeiros Passos com o Arduino**. Editora Novatec. Primeira edição. 2011
- [9] ALLEGRO. **DMOS Microstepping Driver with Translator And Overcurrent Protection**. Disponível em: <https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf>
- [10] POLOLU. **A4988 Stepper Motor Drive Carrier**. Disponível em: <<https://www.pololu.com/product/1182>>