



Universidade Estadual de Campinas - Unicamp

Relatório referente à disciplina F530 – Instrumentação I

Automação no processo de deposição de filmes nanoestruturados via adsorção física



Orientador: Prof. Dr. Antonio Riul Junior
riul x(arroba)x ifi.unicamp.br

Co-orinetador: Prof. Dr. Varlei Rodrigues
varlei x(arroba)x ifi.unicamp.br

Cordenador: Prof. Dr. Jose Joaquin Lunazzi

Aluno: Renan Ramkrapes Falcão
renan.ramkrapes x(arroba)x gmail.com

Palavras-chave: *Layer-by-Layer technique, self-healing, superhidrophobic*

Introdução

Muitos foram os desenvolvimentos de tecnologias que aperfeiçoaram a fabricação de filmes nanoestruturados em camadas. Várias são as técnicas que podem ser utilizadas na construção de camadas – imersão, rotação, spray, eletromagnética – mas a técnica de automontagem por adsorção física (LbL, do inglês *Layer-by-Layer*), ilustrada na Figura 1, permite a fabricação de filmes ultrafinos em uma vasta quantidade de materiais com aplicações em óptica, energia, membranas, e biomedicina⁽¹⁾. Neste trabalho, realizaremos a construção de uma braço mecânico que controla a imersão e retirada de lâminas de vidro para fabricação de filmes LbL em dois polieletrólitos de interesse, tecnologia representada no primeiro exemplo da figura abaixo (*Immersive*).

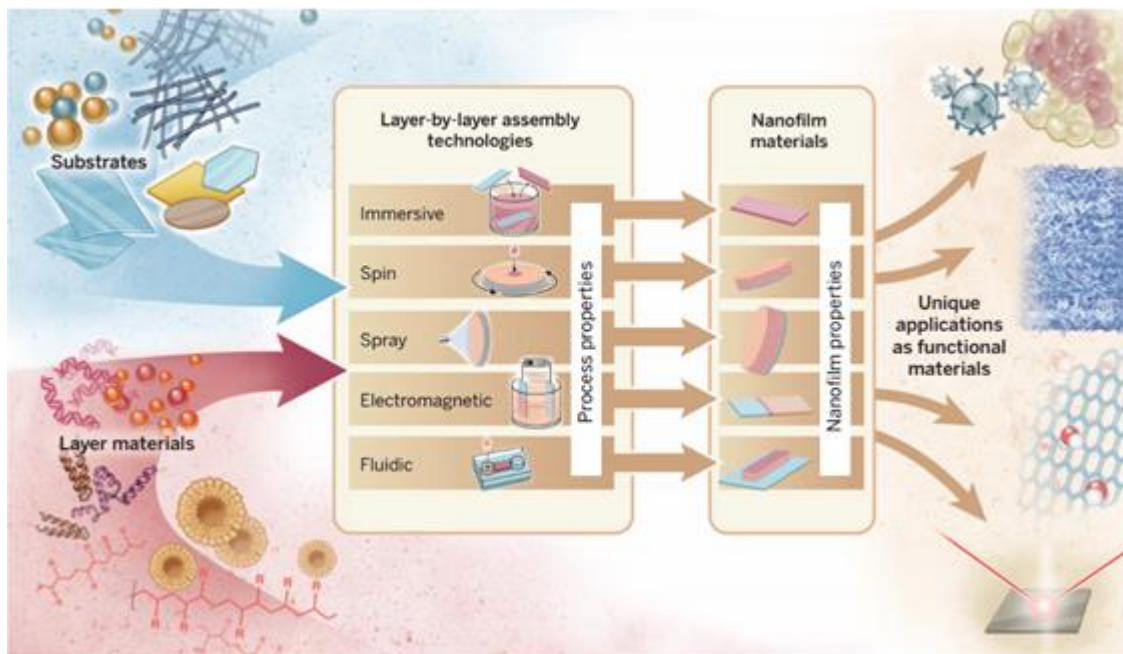


Figura 1 - Diferentes tecnologias de montagem *Layer-by-Layer*. Adaptada de Science 24 Apr 2015: Vol. 348, pags. 411-423; 2015⁽¹⁾.

Os polieletrólitos utilizados na de filmes LbL visam aplicações no dia a dia que vão desde criar materiais hiperhidrofóbicos até aumentar sua durabilidade por meio de auto-regeneração⁽²⁾. A importância da automação do sistema envolvido neste projeto se dá pela necessidade de melhorar reprodutibilidade de resultados do processo de fabricação dos filmes LbL, quando comparado à metodologia manual.

Objetivo específico

O objetivo deste projeto é a automação de um sistema de imersão para fabricação de filmes LbL nanoestruturados via adsorção física. Será montado um braço mecânico responsável pela imersão dos filmes que realizará um movimento vertical preciso com controle de posição, velocidade de subida e descida e tempo de parada, bem como a rotação para escolha dos líquidos de cada recipiente. Dessa forma, é possível minimizar deformações na montagem LbL, ilustrados Figura 2, garantindo a suspensão estável do filme nos processos de imersão e secagem.

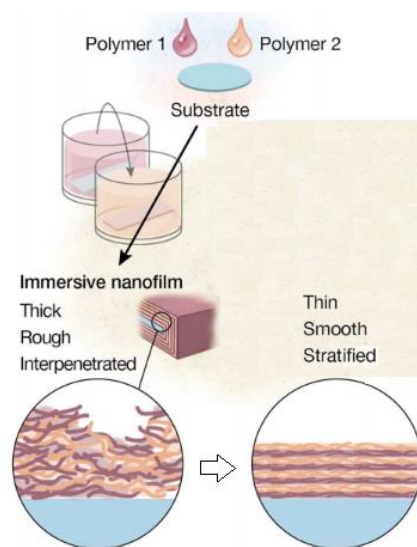


Figura 2 – Processo de imersão manual e o resultado esperado com o braço mecânico ⁽¹⁾

Objetos de Estudo

Durante a montagem de todo o sistema de automação os principais objetos utilizados são uma impressora 3D,⁽³⁾ responsável pela confecção das peças em polímeros, a plataforma de *hardware* e *software* livre Arduino⁽⁵⁾ que, juntamente ao driver A4988⁽⁸⁾, controla os parâmetros de dois motores de passo⁽⁹⁾ que serão responsáveis pelo movimento do sistema.

1. Impressora 3D

Apesar de estar mais popularizada nos dias atuais, a impressora 3D tem uma história que remonta aos anos 80⁽³⁾. Uma das primeiras tecnologias de impressão foi a montagem camada por camada, conforme ilustrado na Figura 3, que será a

mesma utilizada para confecção das peças desse projeto. O processo de confecção das peças se dá por meio da técnica *FDM* (do inglês *Fused Deposition Modelling*) que é a deposição de filamentos termoplásticos amolecidos criando o número de camadas desejadas⁽⁴⁾.



Figura 3 – Ilustração da confecção de peças por camada⁽⁴⁾.

Para criação dos desenhos 3D foi utilizado um software livre de CAD (do inglês *Computer Aided Design*) chamado de OpenScad⁽¹⁰⁾. Porém, para construção do projeto utilizaremos as figuras já desenhadas e exportadas na extensão *.scad*, para que seja possível abrir em um outro *software* livre chamado Slic3r⁽¹¹⁾.

O Slic3r é o programa responsável por transformar o objeto 3D em uma linguagem que a impressora entenda (GCode⁽¹¹⁾), definindo velocidade, altura das camadas, total de preenchimento da peça impressa, perímetros, diâmetro do filamento, etc. Após a criação do GCode é necessário transferir esse arquivo para a impressora. Para isto é utilizado outro *software* livre, o Pronterface⁽¹²⁾.

2. Arduino

O Arduino representado na Figura 4 é uma plataforma de hardware e software livre que tem sido o cérebro de milhares de projetos, desde objetos do cotidiano, por ser de fácil entendimento, a instrumentos científicos complexos.⁽⁵⁾ Utilizando essencialmente a linguagem de programação C/C++, seus sistemas digitais se comunicam com dispositivos eletrônicos externos e possibilitam o controle dos mesmos de acordo com o código de execução criados no IDE Arduino.



Figura 4 – Visualização do hardware do Arduino UNO⁽⁶⁾.

3. Driver A4988

Para o controle dos motores, a placa do Arduino não seria suficiente para criar uma rotina de deposição de filmes. Assim, foram utilizados dois drivers A4988 que foram concebidos especificamente para auxílio no controle de motores de passo, conforme Figura 5.

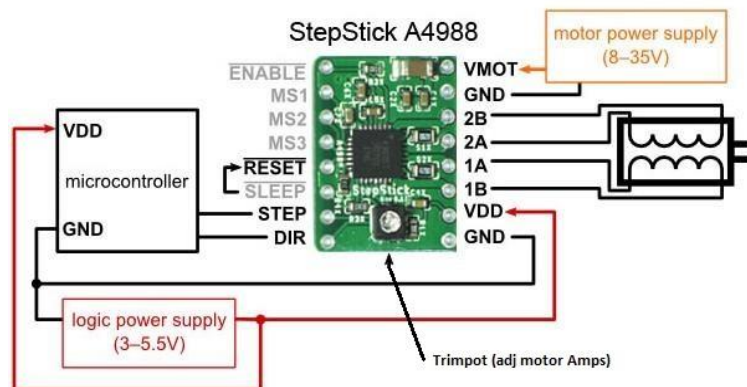


Figura 5 – Visualização do driver A4988 e as ligações que serão realizadas⁽⁸⁾.

Como é um controlador de motor de passo, é de se esperar que tenha entradas e saídas específicas. Percebem-se na Figura 5 as saídas para as bobinas dos motores, uma alimentação auxiliar de uma fonte externa, alimentação direta do Arduino, pinos de controle do sentido, habilitação e passos do movimento, além das entradas que determinam quantos passos o motor dará por volta, as chamadas MS1, MS2 e MS3.

Montagem do Projeto

Toda a concepção do projeto se dá na criação de camadas nanoestruturadas nos filmes. Dessa forma, é necessário que mergulhemos o objeto desejado nos líquidos de interesse, controlando tempo de secagem e lavagem em água. Para tal montagem, foi proposto um movimento vertical para efetuar o mergulho, bem como movimento rotacional para deposição em 3 béqueres diferentes (dois líquidos de interesse e água para lavagem), utilizando dois motores de passo.

1. Estrutura Física

A Figura 6 contém a impressora utilizada para testes e impressões de peças utilizadas neste trabalho.

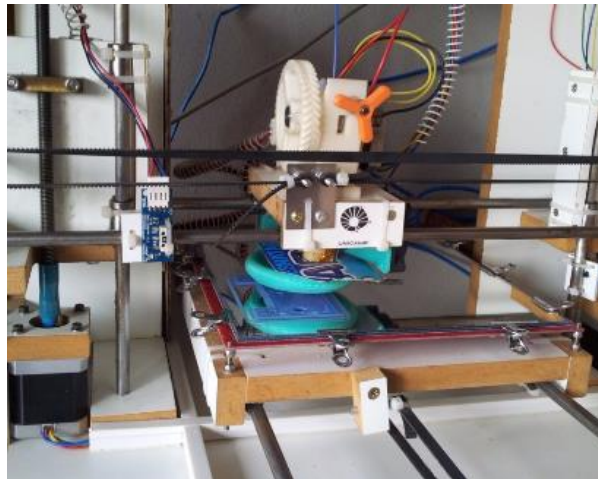


Figura 6 – Impressora utilizada para impressão das peças

As figuras 8-11 a seguir comparam os desenhos em CAD com os resultados impressos. São necessários um suporte para o motor que gira uma das barras roscadas e gera o movimento vertical; um suporte superior para alinhamento; suporte para o rolamento e para um segundo motor responsável pela rotação; e um apoio de fixação para essa peça. Por último, a pinça responsável por fixar outra barra roscada e efetivamente mergulhar o filme no béquer.

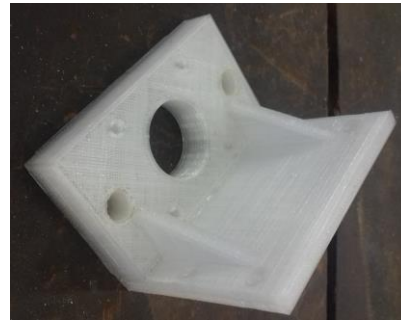
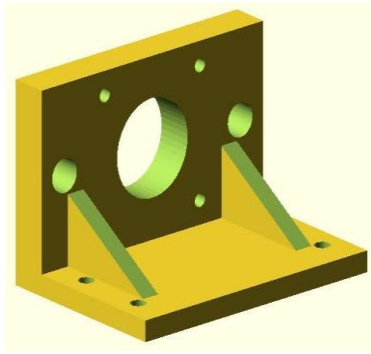


Figura 7 – Peça de suporte inferior para o motor de passo do movimento vertical. É possível visualizar o furo do eixo central onde o motor girará uma barra roscada.

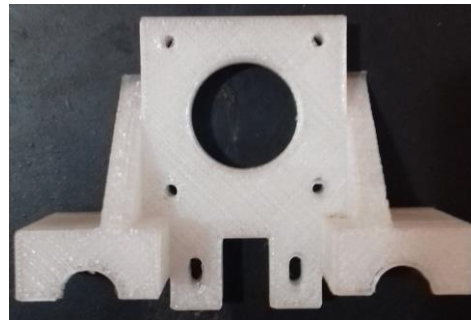
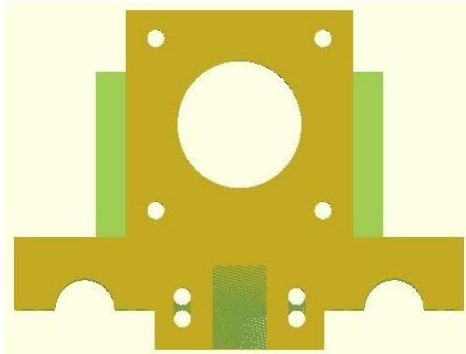


Figura 8 – Peças de suporte para o segundo motor com encaixes no eixo central e nos auxiliares.

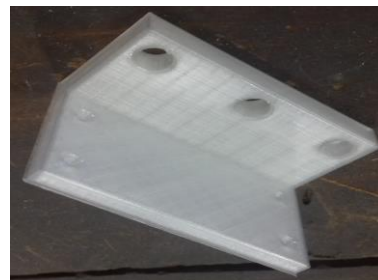
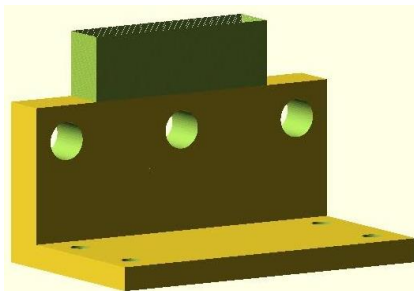


Figura 9 – Suporte superior no lado oposto do motor de passo

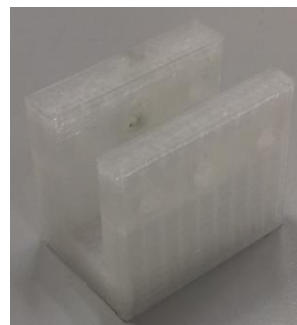
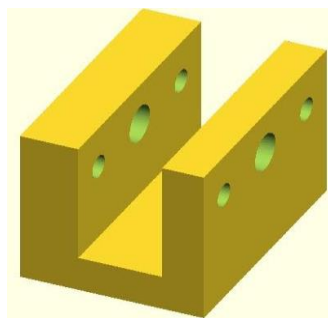


Figura 10 – Suporte de encaixe na peça do rolamento

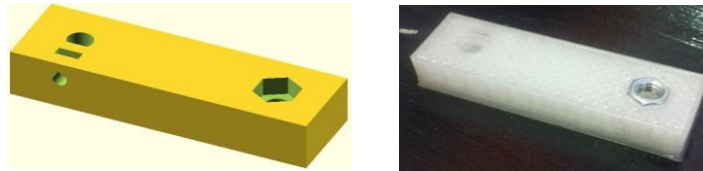


Figura 11 – Pinça que segura a segunda barra roscada e que gira após rotação do motor de escolha dos béqueres.

Para montagem do projeto, além das peças impressas, foi necessário o uso de outros materiais, descritos abaixo:

- Duas barras roscadas 5M (diâmetro de 5mm), 20cm e 12cm;
- Três porcas para barra roscada 5M;
- Duas barras lisas de aço (diâmetro de 8mm) de 20 cm;
- Dois rolamentos com 8mm de diâmetro interno;

Primeiramente, os eixos lisos foram encaixados nos suportes das extremidades com o auxílio de uma morsa, para que ficassem exatamente alinhados, conforme Figura 12. Percebe-se que os dois rolamentos já foram colocados para futuro encaixe do suporte. É necessário que os eixos estejam bem lubrificados para um movimento adequado e para evitar uma precoce evitar oxidação.



Figura 12 – Eixos lisos já colocados nos suportes das extremidades com auxílio de uma morsa para melhor alinhamento.

Em seguida, o motor responsável pelo movimento vertical foi colocado e encaixado na barra roscada de 20 mm para que fosse possível acoplar o suporte do rolamento. A Figura 13 mostra a forma como foi feito o encaixe do suporte do rolamento e dos motores nos eixos do sistema.

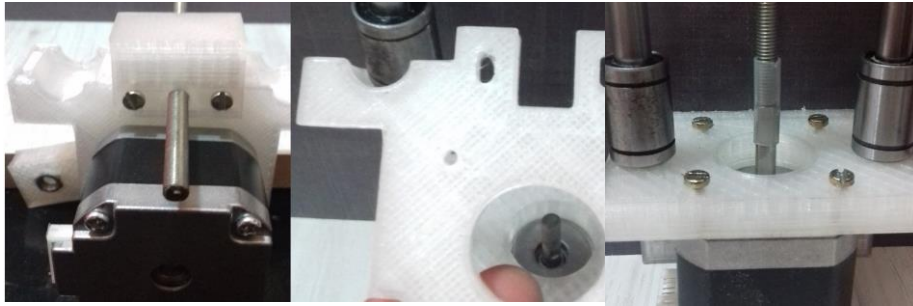


Figura 13 – Esquema de montagem do suporte ao rolamento e motor de rotação e a junção da barra roscada ao motor de movimento vertical.

A montagem final da parte física do projeto se dá ao colocar a pinça com a segunda barra roscada no suporte do rolamento. A Figura 14 ilustra o resultado obtido, faltando agora montar o circuito elétrico e escrever o código em linguagem de programação de acordo com a rotina desejada.

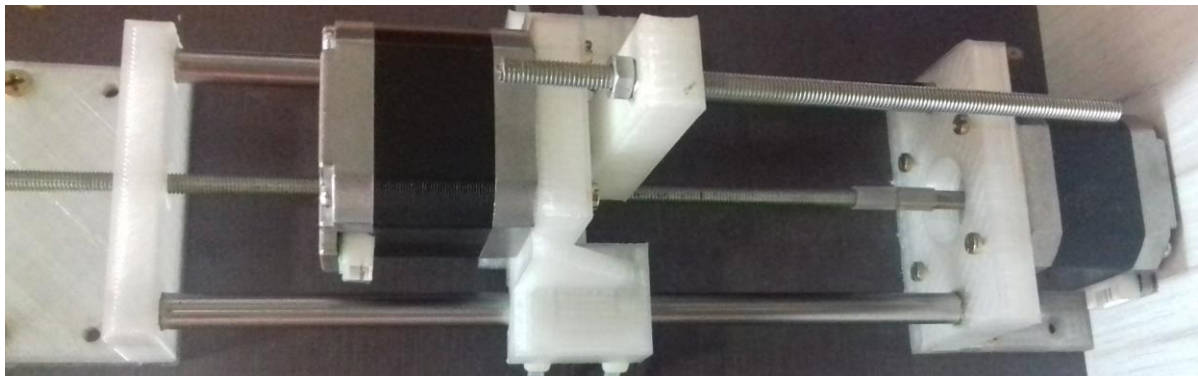


Figura 14 – Foto da parte física do projeto finalizada

2. Circuito Elétrico

Para o projeto, o motor responsável pelo movimento vertical teve seu respectivo driver com ligações para 1600 passos por volta e o motor responsável pela rotação e troca dos béqueres com 3200 passos por volta. Os valores escolhidos se justificam nas mínimas velocidades, pois os movimentos do braço mecânico devem ser o mais suave possível, evitando altas velocidades. Essas configurações podem ser visualizadas na Figura 15, que mostra as ligações do circuito elétrico. Além de definir os passos dos motores, o driver A4988 tem uma chave controladora de corrente, que auxilia no movimento para evitar que o sistema trema, visto que velocidades muito baixas podem causar esse problema que prejudicaria a construção das camadas

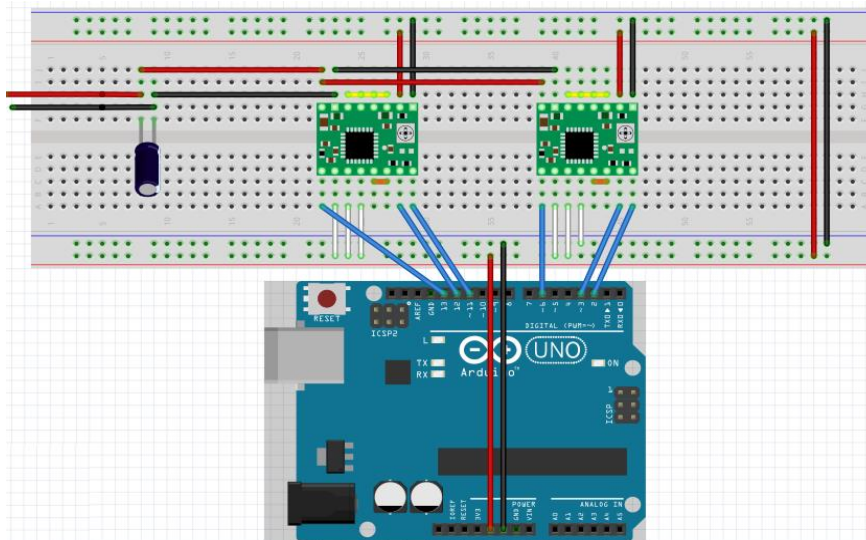


Figura 15 – Esquema do circuito elétrico para as ligações do Arduino e dos drivers.

Na figura percebe-se a utilização de um capacitor de 100 μ F em paralelo à alimentação da fonte externa, que minimiza possíveis alterações de sinal, visto que geralmente ela se dá por um equipamento ligado à rede de distribuição elétrica padrão.

Em amarelo estão as saídas que serão ligadas nas bobinas dos motores. É possível identificar em laranja pequenos jumpers nas entradas dos drivers pois tais funções não serão utilizadas. O que realmente será controlado pelo Arduino serão as entradas de passo, direção e desativação, colocadas na cor azul. Esses serão os parâmetros que sofrerão mudanças no decorrer da rotina.

Finalizado o circuito elétrico, falta apenas a concepção de um código que controle efetivamente o movimento do sistema. Primeiramente, é importante lembrar qual movimento o braço realmente precisa efetuar.

3. Código da Rotina

A Figura 16 representa a visão do movimento que a pinça acoplada ao motor fará.

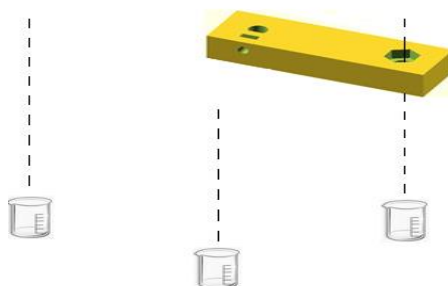


Figura 16 – Visão do movimento de rotação da pinça.

Primeiramente o filme é mergulhado no béquer da esquerda por meio do motor de movimento vertical, onde está contido o primeiro polieletrólito de interesse. Em seguida sobe para secagem e segue para o béquer central contendo água para lavagem. Terminado esse primeiro processo e sua secagem, ele segue para o último béquer contendo outro polieletrólito de interesse, passando novamente pelo béquer central com água para nova lavagem, finalizando o ciclo retornando ao local inicial.

Seguindo essa lógica, será necessária uma velocidade muito baixa para que o motor desenvolva um movimento adequado. O programa será escrito de uma maneira que seja possível escolher o ângulo entre os béqueres da Figura 16.

O movimento vertical é gerado pela rotação de um motor ligado a uma barra roscada, que dá 1600 passos por volta. A barra utilizada tem como deslocamento padrão 0,8 mm/volta. Dessa forma, é possível determinar qual o deslocamento de cada passo, que é 0,5 μm . Essa informação é importante, pois com o tempo de deslocamento de cada passo podemos escolher a velocidade do movimento de acordo com a rotina desejada.

O movimento de rotação para troca dos béqueres tem um comportamento diferente. O outro motor rotaciona menos de meio giro ao total, com ângulo de abertura entre os béqueres de acordo com o desejado pelo usuário.

Além disso, como a ideia do projeto é que ele seja capaz de criar camadas nos filmes, é necessário que os ciclos se repitam de maneira a reproduzir o desenho da Figura 2, com várias camadas de uma rotina bem definida. Com parâmetros básicos do movimento determinados, são acrescentados no código os tempos de imersão nos 3 béqueres para criação precisa da camada.

A velocidade padrão para o deslocamento vertical com bom desempenho na criação das camadas é de 5 mm/min, mas metade desse movimento pode ser feito com o dobro dessa velocidade para diminuir o tempo de rotina. Dessa forma, quando próximo aos béqueres o movimento tem velocidade 5 mm/min e quando mais distante tem velocidade 10 mm/min. O tempo de imersão varia para cada polieletrólito e a forma com que é criada a monocamada nanoestruturada adsorvida.

Dentro dos parâmetros acima citados, o código escrito exige valores de entrada para desenvolver a rotina. São eles:

- Número de ciclos;
- Velocidade vertical em milímetros por minuto;
- Deslocamento vertical em milímetros;
- Ângulo entre os béqueres em graus;
- Tempos de imersão em cada béquer.

No Anexo I desse relatório está contido todo o código escrito para criação da rotina.

Na Figura 17 está o projeto finalizado, juntamente a um suporte de madeira em forma de “L”, feito em marcenaria, para melhor alocação e alinhamento.

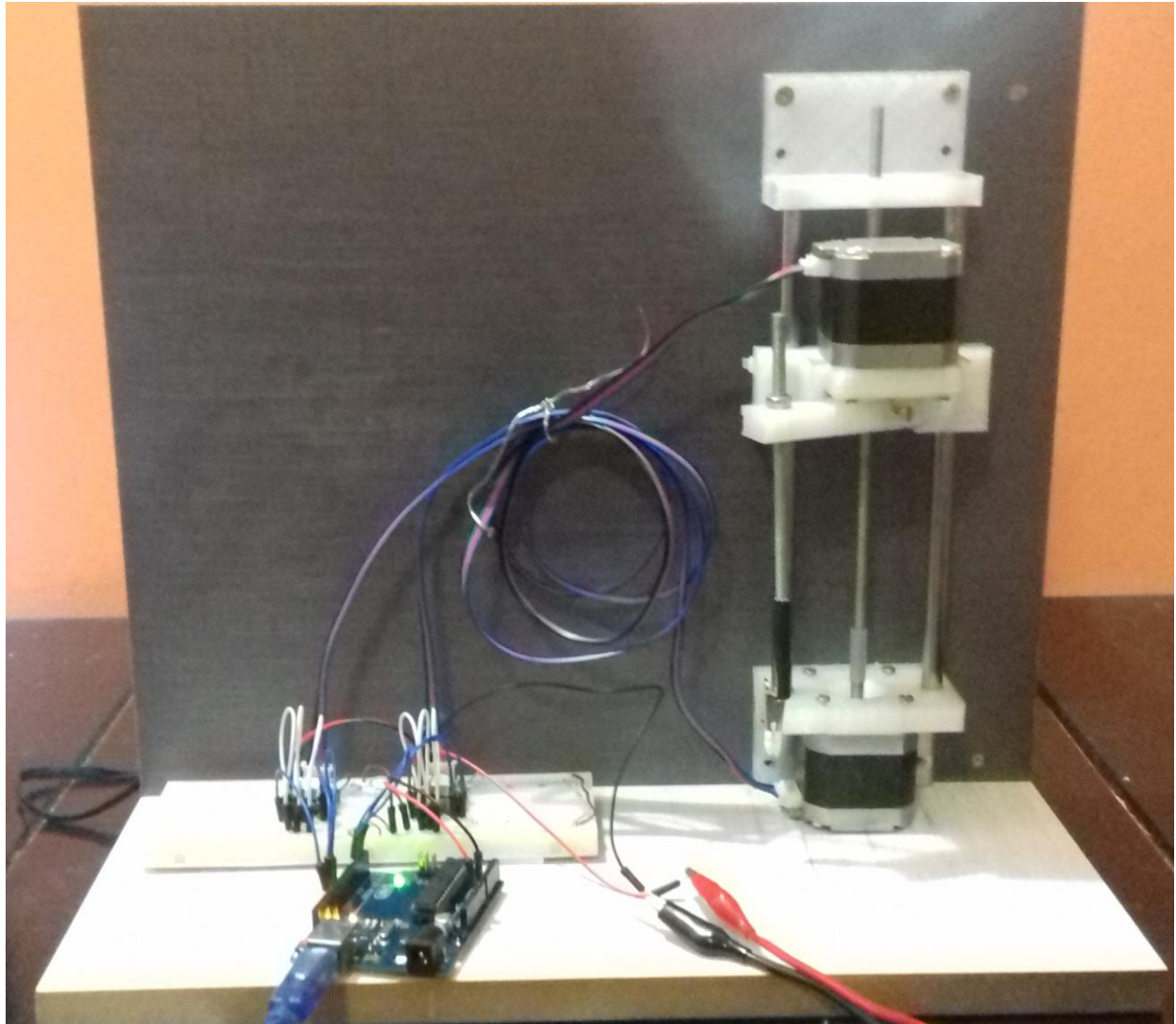


Figura 17 – Projeto final montado e em pleno funcionamento.

Conclusão

O desenvolvimento do projeto para automação do processo de deposição de filmes nanoestruturados possibilitou o conhecimento e utilização de impressoras 3D e a plataforma livre Arduino, abrangendo desde sua eletrônica até a linguagem de programação. A concepção do projeto e os movimentos desejados foram fundamentais na construção do braço mecânico, visando a deposição de filmes ultrafinos nanoestruturados de forma precisa.

A automação do processo de deposição nos dá a possibilidade da reprodução dos filmes fiéis à rotina escolhida, sem alterações ou deformações que manualmente certamente ocorreriam. O ponto principal dessa montagem é a baixa velocidade proporcionada pelo esquema, facilitando a adsorção e tempo de secagem das monocamadas, pois com as velocidades envolvidas a lâmina sai praticamente seca dos béqueres dos polieletrólitos.

Interação na consulta à comunidade:

O público que interagiu variou desde uma família até alunos dos laboratórios de ensino que estavam em aula. Tive a oportunidade de conversar sobre outros assuntos ao mesmo tempo em que apresentava o projeto e tirava dúvidas. Fiquei contente com o resultado da apresentação e acho que foi possível explicar a motivação do projeto bem como sua concepção.

Meu orientador concorda com o expressado neste relatório e deu a seguinte opinião:

“O aluno realizou um ótimo trabalho, cumprindo todas as etapas no planejamento proposto. Terminou o projeto com perfeição, valendo a pena destacar tempo limitado que dispunha para realização do projeto por trabalhar durante o dia e estudar no período noturno. Destacamos seu empenho e capacidade para estudos avançados, além de manter ótimo ambiente de trabalho com colegas no grupo de pesquisa.”

Referências

- [1] Technology-driven layer-by-layer assembly of nanofilms: Disponível em <<http://science.sciencemag.org/content/348/6233/aaa2491>> Acesso em 23 de Agosto de 2016
- [2] How Nanotechnology Can Change the Concrete World: Disponível em <<https://pantherfile.uwm.edu/sobolev/www/SobolevP1.pdf>> Acesso em 23 de Agosto de 2016
- [3] A Brief History of 3D Printing: Disponível em <https://individual.troweprice.com/staticFiles/Retail/Shared/PDFs/3D_Printing_Infographic_FINAL.pdf> Acesso em 05 de Outubro de 2016
- [4] Fused Deposition Modeling: Most Common 3D Printing Method: Disponível em <<http://www.livescience.com/39810-fused-deposition-modeling.html>> Acesso em 05 de Outubro de 2016
- [5] What is Arduino?: Disponível em <<https://www.arduino.cc/en/Guide/Introduction>> Acesso em 05 de Outubro de 2016
- [6] Arduino UNO & Genuino UNO: Disponível em <<https://www.arduino.cc/en/Main/ArduinoBoardUno>> Acesso em 05 de Outubro de 2016
- [7] Controle de motor de passo bipolar com o driver A4988: Disponível em <<http://www.arduinoocia.com.br/2015/03/controle-motor-de-passo-bipolar-driver-A4988.html>> Acesso em 14 de Outubro de 2016
- [8] A4988 – Driver de Motor de Passo: Disponível em <<http://omecatronico.com.br/blog/a4988-driver-de-motor-de-passo/>> Acesso em 14 de Outubro de 2016
- [9] Brites, F. G.; Santos, V. P. A.; Motor de Passo, Niterói, 2008: <<https://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>> Acesso em 14 de Outubro de 2016
- [10] About OpenSCAD: Disponível em <<http://www.openscad.org/about.html>> Acesso em 14 de Outubro de 2016
- [11] About Slic3r: Disponível em <<http://slic3r.org/about>> Acesso em 14 de Outubro de 2016
- [12] About Pronterface: Disponível em <<http://www.pronterface.com/>> Acesso em 14 de Outubro de 2016

Anexo I

```
// ***** CODIGO CONTROLE DE BRACO MECANICO PARA DEPOSICAO DE FILMES NANOESTRUTURADOS ***** //
```

```
int n = 1; // numero de ciclos desejados
float vel_vert = 10; // velocidade vertical em mm/min
float desl_vert = 100; // deslocamento em milímetros
float angulo = 72; // angulo em graus entre os recipientes

int t_imerso_liq1 = 3000; // tempo imerso no liquido 1 em milisegundos
int t_imerso_liq2 = 3000; // tempo imerso no liquido 2 em milisegundos
int t_imerso_agua = 3000; // tempo imerso na agua em milisegundos

float desl_passo_vert = 5E-4; // deslocamento em mm de um passo do motor vertical 1/8 (0,5 micrometros)
float desl_passo_giro = 25E-5; // deslocamento em mm de um passo do motor de giro 1/16 (0,25 micrometros)

float t_passo_vert = (desl_passo_vert/vel_vert)*(3E7); // tempo de cada passo em deslocamento vertical em microsegundos (3 e nao 1)
float t_passo_giro = 30000; // tempo de cada passo em deslocamento de giro em microsegundos

double voltas = (desl_vert/0.8); // numero de voltas do motor que do movimento vertical escolhido
int giro = (3200*angulo)/360; // passos para completar o angulo de giro desejado

const int stepPin1 = 3;
const int dirPin1 = 2;
const int enaPin1 = 4;
const int stepPin2 = 12;
const int dirPin2 = 11;
const int enaPin2 = 13;

int i, j, k=0;

void setup() {

    pinMode(stepPin1, OUTPUT);
    pinMode(dirPin1, OUTPUT);
    pinMode(enaPin1, OUTPUT);
    pinMode(stepPin2, OUTPUT);
    pinMode(dirPin2, OUTPUT);
    pinMode(enaPin2, OUTPUT);

    digitalWrite(enaPin1, LOW);
    digitalWrite(enaPin2, LOW);
}

void loop() {

    int ponto = voltas/2;

    while(k<n){

        for(i=0; i<voltas; i++){

            // descida para liquido 1

            if(i>ponto) t_passo_vert = 1500; // valor para atingir 5mm/min que é a velocidade de descida e subida próxima ao recip:

            for(j=0; j<1600; j++){

                digitalWrite(dirPin1, LOW);
                digitalWrite(stepPin1, HIGH);
                delayMicroseconds(t_passo_vert);
                digitalWrite(stepPin1, LOW);
                delayMicroseconds(t_passo_vert);
            }
        }
    }
}
```

```

delay(t_imerso_liq1); // tempo de imersão no líquido 1
for(i=0; i<voltas; i++){ // subida
    if(i>ponto) t_passo_vert = (desl_passo_vert/vel_vert)*(3E7);
    for(j=0; j<1600; j++){
        digitalWrite(dirPin1, HIGH);
        digitalWrite(stepPin1, HIGH);
        delayMicroseconds(t_passo_vert);
        digitalWrite(stepPin1, LOW);
        delayMicroseconds(t_passo_vert);
    }
for(i=0; i<giro; i++){ // giro para o bequer central
    digitalWrite(dirPin2, HIGH);
    digitalWrite(stepPin2, HIGH);
    delayMicroseconds(t_passo_giro);
    digitalWrite(stepPin2, LOW);
    delayMicroseconds(t_passo_giro);
}
for(i=0; i<voltas; i++){ // descida para agua
    if(i>ponto) t_passo_vert = 1500;
    for(j=0; j<1600; j++){
        digitalWrite(dirPin1, LOW);
        digitalWrite(stepPin1, HIGH);
        delayMicroseconds(t_passo_vert);
        digitalWrite(stepPin1, LOW);
        delayMicroseconds(t_passo_vert);
    }
}
delay(t_imerso_agua); // tempo de imersao na agua
for(i=0; i<voltas; i++){ // subida
    if(i>ponto) t_passo_vert = (desl_passo_vert/vel_vert)*(3E7);
    for(j=0; j<1600; j++){
        digitalWrite(dirPin1, HIGH);
        digitalWrite(stepPin1, HIGH);
        delayMicroseconds(t_passo_vert);
        digitalWrite(stepPin1, LOW);
        delayMicroseconds(t_passo_vert);
    }
}
for(i=0; i<giro; i++){ // giro para o ultimo bequer
    digitalWrite(dirPin2, HIGH);
    digitalWrite(stepPin2, HIGH);
    delayMicroseconds(t_passo_giro);
    digitalWrite(stepPin2, LOW);
    delayMicroseconds(t_passo_giro);
}

```

```

for(i=0; i<voltas; i++){ // descida para o liquido 2

    if(i>ponto) t_passo_vert = 1500;

    for(j=0; j<1600; j++){

        digitalWrite(dirPin1, LOW);
        digitalWrite(stepPin1, HIGH);
        delayMicroseconds(t_passo_vert);
        digitalWrite(stepPin1, LOW);
        delayMicroseconds(t_passo_vert);
    }
}

delay(t_imerso_liq2); // tempo de imersao no liquido 2

for(i=0; i<voltas; i++){ // subida

    if(i>ponto) t_passo_vert = (desl_passo_vert/vel_vert)*(3E7);

    for(j=0; j<1600; j++){

        digitalWrite(dirPin1, HIGH);
        digitalWrite(stepPin1, HIGH);
        delayMicroseconds(t_passo_vert);
        digitalWrite(stepPin1, LOW);
        delayMicroseconds(t_passo_vert);
    }
}

for(i=0; i<giro; i++){ // giro para retornar ao bequer central

for(i=0; i<voltas; i++){ // descida para a agua

    if(i>ponto) t_passo_vert = 1500;

    for(j=0; j<1600; j++){

        digitalWrite(dirPin1, LOW);
        digitalWrite(stepPin1, HIGH);
        delayMicroseconds(t_passo_vert);
        digitalWrite(stepPin1, LOW);
        delayMicroseconds(t_passo_vert);
    }
}

delay(t_imerso_agua); // tempo de imersao na agua

for(i=0; i<voltas; i++){ // subida

    if(i>ponto) t_passo_vert = (desl_passo_vert/vel_vert)*(3E7);

    for(j=0; j<1600; j++){

        digitalWrite(dirPin1, HIGH);
        digitalWrite(stepPin1, HIGH);
        delayMicroseconds(t_passo_vert);
        digitalWrite(stepPin1, LOW);
        delayMicroseconds(t_passo_vert);
    }
}

for(i=0; i<giro; i++){ // giro para retornar ao bequer inicial

    digitalWrite(dirPin2, LOW);
    digitalWrite(stepPin2, HIGH);
    delayMicroseconds(t_passo_giro);
    digitalWrite(stepPin2, LOW);
    delayMicroseconds(t_passo_giro);
}
k++;

if(k==n){ // comando que determina o fim da rotina ao atingir o numero de ciclos

    digitalWrite(enaPin1, HIGH);
    digitalWrite(enaPin2, HIGH);
}
}
}

```