UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

FERNANDO WESLLEY SILVA DE OLVEIRA

AUTOMAÇÃO DE SISTEMA DE MOTORES DE UM FORNO ELÉTRICO PARA DEPOSIÇÃO DE FILMES FINOS PELA TÉCNICA DE SPRAY-PIRÓLISE

FORTALEZA
2020

FERNANDO WESLLEY SILVA DE OLIVEIRA

AUTOMAÇÃO DE SISTEMA DE MOTORES DE UM FORNO ELÉTRICO PARA DEPOSIÇÃO DE FILMES FINOS PELA TÉCNICA DE SPRAY-PIRÓLISE

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Engenharia Mecânica. Área de concentração: Processos Equipamentos e Sistemas para Energias Renováveis. Linha de Pesquisa: Energia Solar (Térmica e Fotovoltaica).

Orientador: Prof. Dr. Manuel Pedro Fernandes Graça.
Coorientadora: Profª. Dra. Ana Fabíola Leite Almeida.

FORTALEZA

2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

O47a Oliveira, Fernando Weslley Silva de.
Automação de sistema de motores de um forno elétrico para deposição de filmes finos pela técnica de spray-pirólise / Fernando Weslley Silva de Oliveira. – 2020.
90 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Mecânica, Fortaleza, 2020.
Orientação: Prof. Dr. Manuel Pedro Fernandes Graça.
Coorientação: Profa. Dra. Ana Fabíola Leite Almeida.

1. Programação Computacional. 2. Deposição em Substrato. 3. Câmara de Aquecimento. 4. Processos Automatizados. 5. Produção de Células Solares. I. Título.

CDD 620.1

FERNANDO WESLLEY SILVA DE OLIVEIRA

AUTOMAÇÃO DE SISTEMA DE MOTORES DE UM FORNO ELÉTRICO PARA DEPOSIÇÃO DE FILMES FINOS PELA TÉCNICA DE SPRAY-PIRÓLISE

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica, do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial para a obtenção do Título de Mestre em Engenharia Mecânica. Área de Concentração: Processos, Equipamentos e Sistemas para Energias Renováveis.

Aprovada em 27/04/2020

BANCA EXAMINADORA

_____

Prof. Dr. Manuel Pedro Fernandes Graça (Orientador)
Universidade de Aveiro (UA)

_____

Prof.ª Dra. Ana Fabíola Leite Almeida (Coorientadora)
Universidade Federal do Ceará (UFC)

_____

Prof.ª Dra. Maria Alexsandra de Sousa Rios
Universidade Federal do Ceará (UFC)

_____

Prof. Dr. Ivan Jeferson Sampaio Diogo
Instituto Federal da Paraíba (IFPB)

A Deus.

Aos meus pais, Sandra Mara e Adil.

AGRADECIMENTOS

A Deus, pois sempre recorri a Ele, buncando e conseguindo calma e tranquilidade, nos momentos de tensão, apreensão, ansiedade e também em momentos felizes.

A minha mãe, Sandra Mara, pelo companheirismo e por sempre torcer por mim.

Ao meu pai, Adil, por me ajudar de muitas formas.

A minha irmã, Ilmara, pelo exemplo das conquistas e por correr atrás de seus sonhos.

A minha avó, Nilsa Lopes, que com todo amor sempre permaneceu preocupada comigo.

A todos meus amigos que acompanharam minha luta no mestrado, principalmente Alexandre Holanda, sendo muito companheiro e parceiro e não me deixando abater pelas dificuldades e Ivan, que me incentivou e me deu força e me encorajou.

Ao Mestre Paulo Herbert França, do LAFFER, por todas as dicas e orientações.

A FUNCAP, pelo suporte financeiro.

À professora Ana Fabíola, pela coorientação.

Ao professor Manuel Graça, meu orientador, por acreditar em mim.

Aos demais professores da UFC que se dedicam em transformar nossa Universidade Federal em uma instituição referência e amigável e propícia ao desenvolvimento pessoal, intelectual e profissional dos estudantes.

Aos integrantes da banca, por disporem de seu tempo.

Aos companheiros do Mestrado, pelas reflexões, críticas e sugestões recebidas e também pelos momentos de descontração, confraternização e de muito estudo.

Ao departamento de Engenharia Mecânica, seus integrantes e em especial ao Valdi, secretário da pós-graduação, por ser um exemplo de profissional, sempre proativo e solícito.

“Nossas virtudes e nossos sentimentos são inseparáveis, assim como força e matéria. Quando se separam, o homem dexa de existir.” (TESLA, Nicola)

RESUMO

A automação visa a segurança das pessoas, a qualidade dos produtos, a rapidez da produção e a redução de custos, assim melhorando os complexos objetivos das indústrias e dos serviços. Os processos automatizados crescem na indústria e sua utilização com as energias renováveis está em ascensão, maximizando o potencial de conversão das fontes em energias secundárias e eletricidade. Este trabalho propôs automatizar uma câmara de aquecimento e forno elétrico para obtenção de superfície condutora por meio de um substrato transparente através da implementação de códigos computacionais na IDE do Arduino e uso de linguagens como C, C++ e Python para controlar motores de passo, termopares, drives e *shields*, e realizando a detecção e leitura de temperatura por meio de sensores e atuadores. Foi realizada análise mecânica para visualização das engrenagens e posicionamentos dos componentes; análise elétrica com esquemas e montagens do motor de passo com o arduino; e análise computacional para a elaboração dos algoritmos. Obteve-se códigos que comandaram três motores de passo; um motor M1 responsável por abrir e fechar a tampa do forno elétrico com códigos em C, C++ e Python; um motor M2 responsável pelo movimento no eixo x dentro e no alto da câmara; e um motor M3 para movimentação angular do aerógrafo, de modo que, pela técnica spray pirólise, deposite o líquido no substrato transparente. Houve a implementação de um código para a detecção e leitura da temperatura em dois pontos distintos, utilizando três funções para leitura de protocolo de dados serial SPI, outra para leituras advindas do módulo MAX6675 e uma implementando um SPI para leitura e armazenamento de dados em cartão SD. Foi possível uma análise e entendimento prático das melhores configurações de todos os principais elementos construtivos que compõem um forno a resistência elétrica. O código apresentou uma boa comunicação entre os dois motores de passo. A etapa de detecção de temperatura foi concluída com sucesso, captando as temperaturas e salvando os dados. O uso de outras linguagens de programação no circuito da tampa do forno elétrico mostrou-se uma forte ferramenta como um intermediário de comunicação entre o usuário e o sistema embarcado. O estudo mostrou-se viável para implementação na câmara térmica e forno e para referenciar que processos automatizados sejam realizados além de contribuir cientificamente para futuros projetos de otimização de sistemas energéticos.

Palavras-chave: Programação Computacional. Deposição em Substrato. Câmara de Aquecimento. Processos Automatizados. Produção de Células Solares.

ABSTRACT

Automation aims at the safety of people, the quality of products, the speed of production and the reduction of costs, thus improving the complex objectives of industries and services. Automated processes are growing in the industry and their use with renewable energies is on the rise, maximizing the potential for converting sources into secondary energy and electricity. This work proposed to automate a heating chamber and electric oven to obtain a conductive surface through a transparent substrate through the implementation of computational codes in the Arduino IDE and the use of languages such as C, C ++ and Python to control stepper motors, thermocouples, drives and shields, and performing temperature detection and reading by means of sensors and actuators. Mechanical analysis was performed to visualize the gears and positioning of the components; electrical analysis with diagrams and assemblies of the stepper motor with the arduino; and computational analysis for the development of the algorithms. Codes were obtained that controlled three stepper motors; an M1 engine responsible for opening and closing the electric oven lid with codes in C, C ++ and Python; an M2 motor responsible for the movement on the x axis inside and at the top of the chamber; and an M3 motor for angular movement of the airbrush, so that, using the spray pyrolysis technique, deposit the liquid on the transparent substrate. There was the implementation of a code for the detection and reading of the temperature at two different points, using three functions for reading the SPI serial data protocol, another for reading from the MAX6675 module and one implementing a SPI for reading and storing data on a card SD. It was possible an analysis and practical understanding of the best configurations of all the main construction elements that make up an electric resistance furnace. The code showed good communication between the two stepper motors. The temperature detection step was successfully completed, capturing temperatures and saving data. The use of other programming languages in the electric oven cover circuit proved to be a strong tool as a communication intermediary between the user and the embedded system. The study proved to be feasible for implementation in the thermal chamber and oven and to reference that automated processes are carried out in addition to contributing scientifically to future energy system optimization projects.

Keywords: Computational Programming. Substrate deposition. Heating Chamber. Automated Processes. Solar Cell Production.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

ANSI	*American National Standards Institute*
Ag	Prata
Aneel	Agência Nacional de Energia Elétrica
CdTe	Telureto de Cádmio
CI	Circuito Integrado
CIGS	Disseleneto de Cobre, Índio e Gálio
CLPs	Controladores Lógicos Programáveis
CPT	Condições Padrão de Teste
CSNS	Célula Solar Nanocristalina Sensibilizada por Corante
$Cu(InGa)Se_2$	Seleneto de Cobre e Índio e Gálio
DSSC	*Dye Sensitized Solar Cell*
FAT	*File Allocation Table*
FEM	Força Eletromotriz
GaAs	Arsenieto de Gálio
GaInP	Fosfeto de Índio e Gálio
HCl	Ácido Clorídrico
Icc	Corrente de curto circuito
IDE	*Integrated Development Environment* ou Ambiente Integral de Desenvolvimento
IoT	(Internet das Coisas) *Internet of Things*
ISO	*International Standards Organization*
LAFFER	Laboratório em Filmes Finos em Energias Renováveis
LCD	*Liquid Crystal Display*
LED	*Light Emitting Diode*
M2M	Máquina a Máquina
MISO	*Master IN Slave OUT*
MOSI	*Master OUT Slave IN*
MSG	Método Sol-Gel
NA	Normalmente Aberto
NF	Normalmente Fechado
NH_4F	Fluoreto de Amônio
PC	(Computador Pessoal) *Personal Computer*

PV	*Photovoltaic*
PWM	(Modulação por Largura de Pulso) *Pulse Width Modulation*
RTD	*Resistance Temperature Detector*
SCK	*Serial Clock*
SD	*Secure Digital*
SDHC	*SD High Capacity*
Si	Silício
$SnCl_2.2H_2O$	Cloreto de Estanho II
SnO_2:F	Óxido de Estanho dopado com Flúor
SPA	Técnica de Spray Pirólise
SPI	*Serial Peripheral Interface*
SS	*Slave Select*
TCO (ou OTC)	Óxido Transparente Condutor
TFPV	*Thin-Film Photovoltaic*
TiO_2	Óxido de Titânio
UART	*Universal Asynchronous Receiver / Transmitter*
USB	Universal Serial Bus
Vca	Voltagem de circuito aberto

SUMÁRIO

1 INTRODUÇÃO

A revolução industrial é um marco da evolução fabril e dela surgiu a necessidade de ter um controle centralizado, flexível, barato e que fosse automático (LUGLI e SANTOS, 2019). A automação pode ocorrer em uma máquina que funcionava manualmente ou com grande influência da participação humana. Máquinas menores passaram a ter *softwares*, telas de comando e outros recursos, comunicando-se, formando grandes linhas de montagem completas.

Os sistemas de produção automatizados executam operações tais como processamento, montagem, inspeção e gerenciamento de materiais e são denominados automatizados porque executam suas operações com um nível reduzido de participação humana se comparado ao processo manual equivalente. Em alguns sistemas altamente automatizados, quase não existe participação humana (GROOVER, 2011).

A automação, como qualquer sistema apoiado em computadores, substitui o trabalho humano em favor da segurança das pessoas, da qualidade dos produtos, e da rapidez da produção ou da redução de custos, desta forma, melhorando os complexos objetivos das indústrias e dos serviços (MORAES e CASTRUCCI, 2010). O encadeamento de etapas fica mais ágil e garante uma uniformização de todos objetos que saem desse processo.

Muitos processos têm a necessidade de serem mais precisos em suas aplicações, como a aspersão de um líquido em uma placa, e assim tem-se sempre a mesma quantidade aspergida sobre algum corpo. A técnica do spray pirólise, abordada neste trabalho, utiliza-se deste procedimento.

Há processos cuja finalidade é manter a regularidade e garantia de que a mesma quantidade de material seja depositada em objetos ou placas. A aspersão de substâncias sobre substratos possui a necessidade de ser uniforme, fazendo com que haja uma garantia de igualdade entre elas na etapa final.

As adversidades que são envolvidas nos processos de automação laboratoriais podem variar consideravelmente, conforme os tipos de automação que serão implementados. Pode ser uma automação geral, em que normalmente envolve a aquisição de um dentre os muitos sistemas de automação existentes no mercado; ou ainda ser uma automação pontual, por meio da inserção de ferramentas automáticas em certas etapas do processo laboratorial.

O grau ótimo de automação depende das configurações do laboratório e considerações de custo, volume de atividades, flexibilidade, tempo para instalação, espaço disponível, disponibilidade de pessoal técnico especializado, segurança e confiabilidade (BARBEBO, 2012).

Assim como a necessidade de processos automatizados vem crescendo na indústria, sua utilização em energias renováveis está cada vez mais em evidência. Maximizar o potencial de cada fonte de energia renovável para obtenção da energia elétrica sem grandes danos ao ambiente vem sendo alcançado graças ao avanço tecnológico, sendo a automatização o principal fator propulsor para a melhoria da eficiência dos equipamentos.

Entre as alternativas para produção de energia sustentável e limpa, destaca-se a geração fotovoltaica (FV) - uma fonte de conversão direta da energia solar em energia elétrica. As vantagens da energia elétrica obtida através dos painéis fotovoltaicos estão ligadas à sua estrutura estática, silenciosa, não poluente, renovável e com baixo custo de manutenção, produzindo energia limpa de uma fonte inesgotável (RÜTHER, 2004).

Em países como o Brasil, as células solares de silício possuem a máxima eficiência teórica dentre as demais, apresentando os valores de eficiência de 24,7% para o silício monocristalino, 19,8% para o silício policristalino e 15% para o silício Amorfo; ainda apresentando o disseleneto de cobre, índio e gálio (CIGS) uma eficiência de 18,8% e o Telureto de cádmio (CdTe) de 16,4%. As constantes melhorias nas técnicas de produção têm aumentado a eficiência de conversão das células fotovoltaicas de silício cristalizado, em especial a do silício policristalino, que é o mais utilizado atualmente (DI SOUZA, 2017). Esses valores máximos de eficiência apresentados são todos devido à qualidade da exposição solar (GARCIA, MURAKAMI LHA, 2002). A conversão de energia nas células solares se dá por meio do efeito fotovoltaico e quanto maior for a radiação solar incidindo nos painéis fotovoltaicos, maior será a produção de eletricidade (GRENN, 1982).

Este trabalho, através da automação de sistemas de motores de um forno elétrico, buscou otimizar o processo de deposição de filmes finos pela técnica de spray-pirólise, e o processo de produção das células fotovoltaicas de película fina (TFPV – *Thin-Film Photovoltaic*), que são um procedimento de deposição de uma ou várias camadas finas de material fotovoltaico sobre um substrato, essência básica de como os painéis fotovoltaicos de filme fino são fabricados. (HOFFMANN, PELLKOFER, 2011).

Por conseguinte, é de suma importância a pesquisa por filmes finos, métodos de aquecimento de materiais e ainda por tecnologias que auxiliem todo o processo, deixando-o mais imune a erros, de falhas humanas, aumentando o rendimento e a produtividade.

2 OBJETIVOS

2.1 Objetivo Geral

Automatizar uma câmara de aquecimento e forno elétrico para obtenção de superfície condutora por meio de um substrato transparente (vidro condutor) através da técnica spray pirólise com uso de programação computacional em Arduino, sensores e motores de passo.

2.2 Objetivos Específicos

- Elaborar o código em linguagem Wiring baseada em C/C++ para o Arduino:
    - Controlar motor de passo para a tampa do forno elétrico;
    - Registrar a temperatura do forno através de um termopar.
    - Controlar motor de passo para movimentação do aerógrafo dentro da câmara de aquecimento na direção horizontal e rotacional;
- Elaborar código em linguagem C com a finalidade de proporcionar uma maior usabilidade em outros microcontroladores para automação.
- Implementar parte do código relativo à abertura e ao fechamento da tampa do forno elétrico em linguagem Python e C++.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Energias Renováveis

O desenvolvimento de fontes alternativas de energia, como a energia solar, eólica, biomassa, entre outras, tem recebido cada vez mais destaque nos últimos anos e sendo aprimoradas com estudos, pesquisas e incentivos fiscais, além de vir diversificando as matrizes energéticas dos países. Uma das principais motivações se deve ao aumento da preocupação ambiental, em particular devido às possíveis consequências relacionadas ao efeito estufa para as futuras gerações.

Estima-se que a população mundial se aproximará de 9 bilhões em 2040 e para atender a essa demanda, em comparação ao consumo de 2010, será necessário um aumento próximo a 35% na oferta total de energia. Isto exigirá maior diversificação de recursos energéticos e inovações tecnológicas mais eficientes e confiáveis para o meio ambiente e assim tornar a matriz energética mundial mais sustentável (*International Energy Agency*, IEA, 2019).

O grande impasse do mundo atual é a questão energética, pois o aproveitamento desta ainda não atingiu um nível satisfatório, uma vez que a imensa maioria da energia utilizada no planeta ainda é de origem não renovável. No ano de 2013, cerca de 80% de toda energia produzida foi proveniente da queima de combustíveis fósseis, considerados os vilões causadores do efeito estufa. Esta dependência vem diminuindo nos últimos anos, graças ao uso de fontes renováveis de energia (SAWIN e SVERRISSON, 2019).

A energia solar surge como uma alternativa promissora, principalmente para o Brasil que possui uma posição geográfica privilegiada do planeta. A produção de energia solar fotovoltaica mundial detém apenas 2,4% do mercado de energia renovável, que representa 26,2% do total (SAWIN e SVERRISSON, 2019), dos quais cerca de 90% dos módulos solares comercializados são baseados em células fotovoltaicas de silício mono e policristalino, que possuem elevado custo (SAGA, 2010; GREEN et al, 2012). Desta forma, a baixa representação do mercado fotovoltaico advém do alto custo de fabricação destas células solares, o que representa um entrave para a popularização.

3.2 Energia Solar

Em meio ao advento das fontes renováveis, duas estão em mais destaque no cenário brasileiro: as fontes eólica e solar. O sol apresenta uma fonte inesgotável de energia capaz de

ser transformada em energia elétrica, tanto convertida em energia eólica, como sendo transformada diretamente em módulos fotovoltaicos através da célula fotovoltaica. A terra é aquecida pelo sol de forma irregular – a atmosfera aquece muito mais rapidamente nas regiões equatoriais do que no resto do globo e o solo aquece mais rapidamente do que os oceanos, originando, através de aquecimentos diferenciados da atmosférica, os ventos.

Estima-se que anualmente 3,9x1024 J = 1,08x1018 kWh de energia solar atinge a superfície terrestre, sendo que é mais de 10.000 a demanda anual global por energia e muito maior do que as reservas de energia disponíveis na Terra (QUASCHNING, 2005).

A aplicação de painéis fotovoltaicos está se consolidando no Brasil. No processo fotovoltaico há a geração de energia elétrica por meio da transformação da irradiação solar, chamada de energia solar fotovoltaica. Dentre os principais fatores que influenciam essa geração, há a temperatura do painel e a irradiação solar incidente (SWIEGERS e ENSLIN, 1998). Assim, quanto maior a irradiação solar sobre um painel, maior a corrente elétrica gerada por ele. Todavia, também haverá um aumento na temperatura de operação do módulo, fazendo com que a tensão gerada diminua consideravelmente e, dessa forma, haverá uma diminuição de potência elétrica produzida.

Para conseguir o ponto ótimo dessa conversão, deve-se analisar desempenho elétrico de um módulo fotovoltaico. É fundamental o estudo da curva que relaciona a tensão com a corrente. Esta curva, denominada curva característica I-V apresenta a mesma forma para qualquer painel fotovoltaico nas condições CPT (Condições Padrão de Teste - irradiância de 1000 W/m² e temperatura 25°C) (MEIQIN et al, 2009).

3.3 Energia Solar no Ceará

O crescimento da utilização da Energia Solar no Estado do Ceará, referente à comparação entre 2018 e 2017, é constante, considerando que entre 2017 e 2016 já havia sido registrado uma elevação de mais 98%. Ceará é o atual líder do Nordeste no setor de geração fotovoltaica, segundo dados da Agência Nacional de Energia Elétrica (Aneel).

Embora a maior parte dos sistemas em operação, 71,2%, seja da classe "residencial" (909 unidades geradoras), o segmento "comercial", com 284 unidades, responde por 44% da potência instalada de geração distribuída no Estado, com 9,13 MW, enquanto o residencial dispõe de 5,20 MW. As outras unidades estão distribuídas nas classes "industrial" (33 unidades, e 5,22 MW), "poder público" (21 unidades e 0,96 MW), e "rural" (28 unidades e 0,23 MW), ainda segundo dados da Agência Nacional de Energia Elétrica (Aneel).

Segundo Francisco Duarte, diretor executivo da empresa Ítalo-brasileira G2 Ecoenergia Solar, para 2019, foi previsto para empresa um incremento de cerca de 20% no seu plano de investimento para a instalação de placas fotovoltaicas nas modalidades residencial, comercial e industrial no Ceará. (CABRAL, 2018).

3.4 Filmes Finos

O filme fino ou película, é uma camada de um material que apresenta espessuras geralmente na faixa de frações de nanômetros até 1μm. Os componentes eletrônicos semicondutores e os revestimentos ópticos são os maiores beneficiários do desenvolvimento deste material. Os filmes finos podem ser condutores, semicondutores ou isolantes.

Filmes finos podem ser preparados por varas técnicas como deposição química a vapor (BÉLANGER et al, 1985; MARUYAMA, 1990), deposição metalorgânica, *sputtering*, vaporização, sol-gel (GERALDO et al, 2003) e spray pirólise (THANGARAJU, 2002; ELANGOVAN et al, 2005).

Os filmes finos são a configuração tecnologicamente mais importante, constituindo uma área primordial nas aplicações do Método Sol-Gel (MSG) pelas suas inúmeras utilizações na óptica, na eletrônica e no desenvolvimento de sensores.

Depositar uma ou várias camadas finas de material fotovoltaico sobre um substrato transparente é a essência básica de como os painéis fotovoltaicos de filme fino são fabricados. Células fotovoltaicas de película fina, como são conhecidas, são muito utilizadas dependendo da tecnologia da célula fotovoltaica de filme fino. Os painéis de filme fino possuem eficiências médias entre 7-13%. Algumas tecnologias de painel de filme fino já estão chegando nos 16%, sendo similares a eficiência dos painéis Policristalinos. Em 2015, os painéis fotovoltaicos que utilizam a tecnologia de filme fino representam aproximadamente 20% do mercado mundial de painéis solares fotovoltaicos, sendo a maioria de silício cristalino (SUNERGIA, 2018).

Filmes de TiO_2, sensibilizados por corante de bipiridil-rutênio (GARCIA, 1998, 2002) ou até mesmo corantes orgânicos têm sido vastamente estudados. Os corantes sintéticos possuem alguns problemas, sua síntese é difícil, pois utiliza metais raros em sua composição, por este motivo o uso de corantes orgânicos tem sido muito estudado e incentivado pela comunidade acadêmica. Neste tipo de célula, a luz é absorvida pelo corante e este está ligado quimicamente ao óxido de titânio que foi depositado sobre um Óxido Transparente Condutor (TCO). O corante, ao ser excitado pela luz, ejeta elétrons para a camada do semicondutor.

Normalmente as propriedades do material na forma de filme diferem das propriedades do mesmo material na sua forma maciça devido à influência da superfície. Por outro lado, as propriedades dos filmes são altamente dependentes dos processos de deposição. Um dos aspectos mais interessantes do MSG é o fato do sol fluido, antes de sofrer geleificação, permitir a preparação de filmes finos de uma forma simples e econômica, sendo possível o controle preciso da microestrutura do filme depositado, do volume e tamanho de poro, da espessura e da área superficial.

3.5 Célula de Grätzel

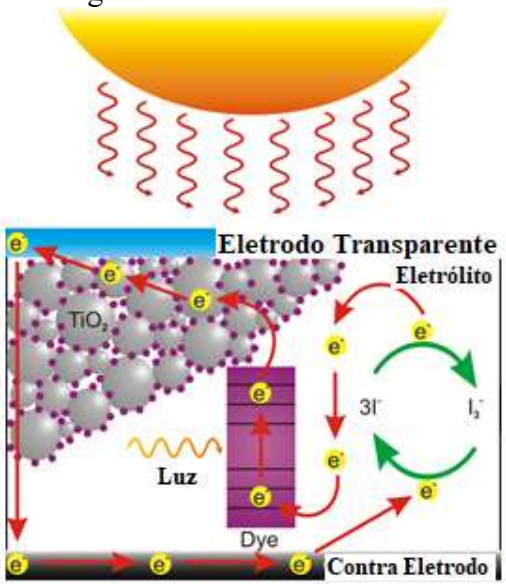
Além dos painéis fotovoltaicos, existem as células solares que também desempenham o papel de converter energia solar em energia elétrica. As chamadas Células Solares Sensibilizadas por Corantes, da sigla DSSC, *dye sensitized solar cell*, são formadas por matérias orgânicos e inorgânicos e são incluídas no grupo de células solares híbridas. Seu sistema de funcionamento é formado por um anodo foto-sensibilizado, baseado em um material semicondutor, um eletrólito e um catodo foto-eletroquímico.

Michel Grätzel e Brian O´Regan apresentaram, na *École Polytechnique Féderale de Lausanne*, em 1991, a proposta de uma célula solar de baixo custo cujo funcionamento estaria embasado no fenômeno da fotossíntese (GRÄTZEL, 1991). A célula de Grätzel (Figura 1), é ensinada nas universidades e é sugerido o uso da mesma, por exemplo, em carregador de dispositivos móveis. As células solares de dióxido de titânio (TiO_2) sensibilizadas por corante surgiram no início da década de 90 e representam uma alternativa interessante para a produção de módulos solares de baixo custo.

As vantagens desse tipo de dispositivo têm procedência por conta de a mistura de matérias orgânicas e inorgânicas serem bastante resistentes, proporcionando uma grande facilidade de processamento, uma vez que as células de silício não possuem tal característica.

As células de Grätzel são mais leves e têm a capacidade se produção em substratos flexíveis, sendo o vidro muito utilizado, sendo assim, estas células possuem grande aplicação, podendo ser coloridas por conta do corante que seja utilizado.

Figura 1 – Células de Grätzel.



Fonte: Adaptado de JONES (2009).

Os vidros condutores são aplicados na construção da célula fotoeletroquímica com eletrodos imersos e da célula de Grätzel. Para isso, além de fazer vidros condutores, é feita a deposição da camada de dióxido de titânio dopado com prata (TiO_2:Ag), preparação de eletrólito, preparação de corante, montagem das células e caracterização por voltagem de circuito aberto (Vca) e corrente de curto circuito (Icc) dentro da câmara de aquecimento.

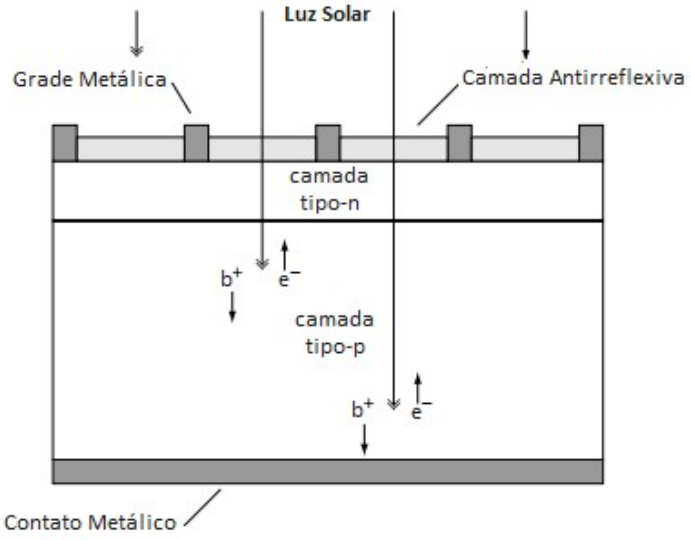
3.6 Estrutura das Células Solares

As células solares semicondutoras são dispositivos simples. Os semicondutores têm a capacidade de absorver a luz e fornecer uma parte da energia dos fótons absorvidos aos transportadores de carga - elétrons e lacunas. Um diodo semicondutor separa e recolhe os transportadores e conduz a corrente elétrica gerada preferencialmente em uma direção específica. Assim, uma célula solar é um diodo semicondutor que foi cuidadosamente projetado e construído para absorver eficientemente e converter energia luminosa do sol em energia elétrica (GRAY, 2003).

Uma estrutura de célula solar convencional simples é representada na figura 2. A luz solar é incidente a partir do topo na parte frontal da célula solar. O diodo semicondutor é formado quando um semicondutor do tipo n e um semicondutor do tipo p são reunidos para formar uma junção metalúrgica. Isto é tipicamente conseguido através da difusão ou implantação de impurezas específicas (dopantes) ou através de um processo de deposição. O

outro contato elétrico do diodo é formado por uma camada metálica na parte de trás da célula solar (GRAY, 2003).

Figura 2 – Estrutura de uma célula solar convencional.



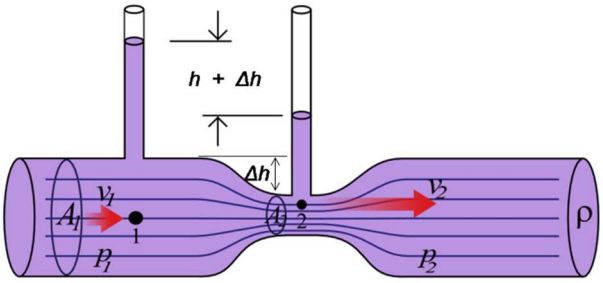
Fonte: Adaptado de GRAY (2003).

As células solares podem ser fabricadas a partir de uma série de materiais semicondutores, sendo o mais comumente utilizado o silício (Si) - cristalino, policristalino e amorfo. Também são fabricadas a partir de GaAs, GaInP, Cu(InGa)Se_2 e CdTe. Os materiais das células solares são escolhidos em grande parte com base no quão bem as suas características de absorção correspondem ao espectro solar e ao seu custo de fabricação (GRAY, 2003).

3.7 Técnica de Spray

Esta técnica, usada há anos, se utiliza de um efeito de diferença de pressão, conhecido como Efeito de Venturi, que nada mais é do que passar ar comprimido (ou outros gases) por um tubo de diâmetro conhecido, sendo este reduzido de forma suave para diminuir os efeitos de turbulência para um diâmetro menor, como mostra a figura 3.

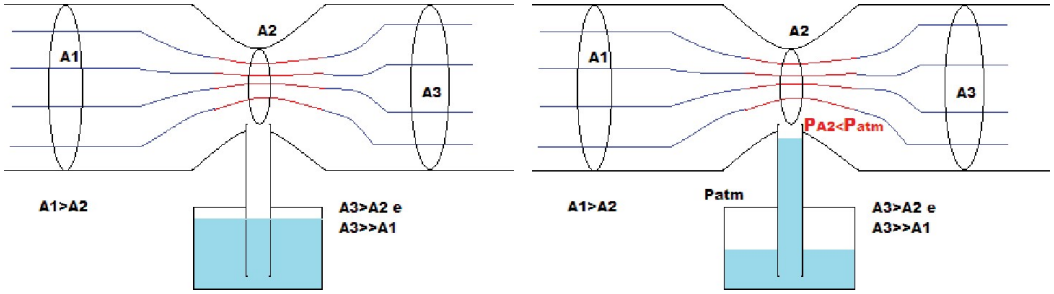
Figura 3 – Tubo de Venturi.



Fonte: CID, A. S., CORREA, T. (2019).

Na área A_1 há uma zona de alta pressão e quando o fluido passa para a área A_2, ele sofre uma redução na pressão e um aumento na velocidade do fluido, entretanto, como neste local há uma zona de baixa pressão, é necessário que a pressão seja compensada. Existe um furo neste local e a pressão é compensada puxando ar através deste furo. Caso haja um fluido de maior densidade abaixo do tubo que entra no tubo de Venturi, este será sugado até o topo do tubo no início da área A_2, conforme a figura 4.

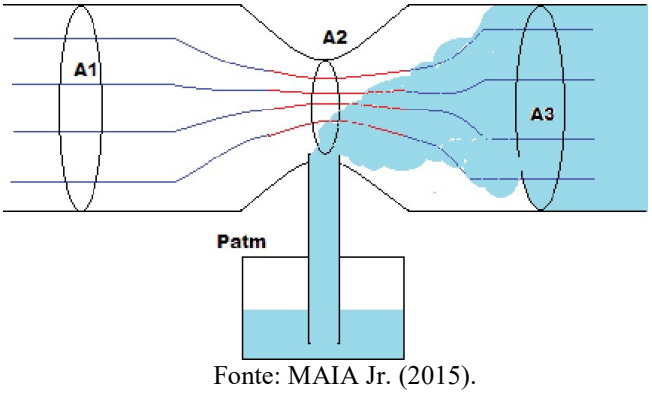
Figura 4 – Tubo de Venturi com Fluxo.



Fonte: MAIA Jr. (2015).

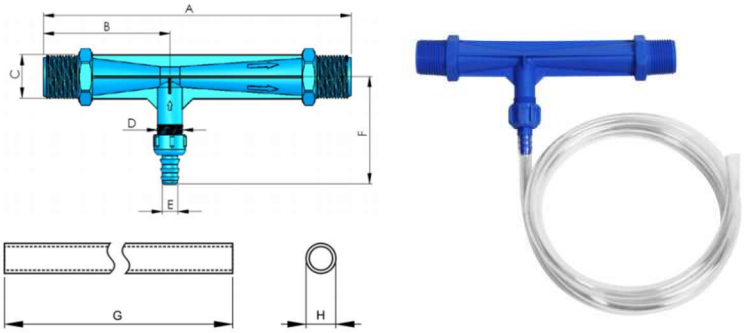
O fluido é aspergido através da área A_3 (Figura 5). De forma exemplificativa, um tubo de Venturi valvulado é ilustrado na figura 6, e injetor altamente eficiente que cria um vácuo com o diferencial de pressão quando a água passa por sua extensão.

Figura 5 – Tubo de Venturi Aspergindo o Fluido.



Fonte: MAIA Jr. (2015).

Figura 6 – Tubo de Venturi Valvulado.



Fonte: FB Irrigação (2020).

3.8 Forno a Resistência

Fornos a resistência utilizam o calor gerado pelo efeito Joule, onde uma corrente elétrica ao percorrer uma determinada resistência produz calor. Tais fornos normalmente não geram oscilações na tensão da rede que os alimentam (CESTILE, 2012).

A maioria dos processos que empregam fornos a resistência requerem uma grande precisão na temperatura da câmara de aquecimento. Em alguns processos a temperatura não pode variar bruscamente, por isso, a temperatura desejada deve ter uma taxa de variação constante até atingir o valor de regime permanente. O sinal de referência deve ser um sinal do tipo rampa até que o forno alcance a temperatura final desejada, quando passa a ser um valor constante (GUERRA, 2006).

Guerra (2006) cita alguns exemplos do emprego de fornos de aquecimento indireto, como o aquecimento de água para produção de vapor, a manutenção da temperatura de fusão do vidro a partir de um bloco de material fundido, fabricação de eletrodos de grafite utilizados em fornos de arco, manutenção do banho que permite a têmpera dos aços, entre outros.

3.9 Spray Pirólise

Técnica utilizada para aplicar filmes finos em substratos, consiste em aquecer o mesmo até uma temperatura adequada de trabalho, e isto varia conforme cada experimento. Uma vez que se tenha atingido a temperatura apropriada, usa-se um uma pistola aerográfica, ilustrada na figura 7, com o auxílio de um compressor, para aspergir a solução sobre o substrato, e esse processo deve ser feito diversas vezes. A figura 8 mostra o kit completo do aerógrafo com o compartimento em que é inserido o líquido a ser aspergido.

Equipamentos mais simples podem ser empregados como borrifador manual (este pode substituir o aerógrafo e o compressor), porém a qualidade da deposição pode ser reduzida, pois nestes equipamentos a aspersão é ineficiente. Normalmente esse processo é feito manualmente, como mostrado na figura 9, o que aumenta o índice de erros e de não conformidade entre os substratos produzidos.

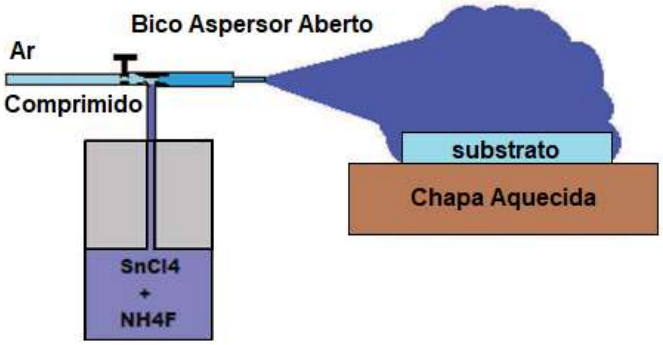
Figura 7 – Aerógrafo utilizado no projeto.

Fonte: Autor.

Figura 8 – Kit Aerógrafo.

Fonte: Autor.

Figura 9 – Processo manual do spray pirólise com pistola aerográfica.



Fonte: MAIA Jr. (2015).

3.10 Arduino

Souza et al. (2011) apresenta o microcontrolador Arduino como uma alternativa aos equipamentos e "kits". Esse microcontrolador exige um investimento bem menor, possui plataforma de desenvolvimento de acesso público e gratuito.

Esses fatores foram de grande peso na escolha dessa plataforma por facilitar o acesso a códigos e muitas explicações sobre diversas aplicações. Um dos fatos que o torna um feito engenhoso na sua criação foi que todos os conectores laterais continuam compatíveis com os do Arduino Uno, disponibilizando a ele todos os *shields* feitos para o UNO (MONK, 2013).

Uma das grandes áreas de atuação do Arduino concentra-se em controle de sistemas, e suas aplicações abrangem campos da robótica, engenharia agronômica, domótica, musical, moda, impressão 3D, engenharia de transportes, indústria, moda, etc (QUEIROZ, 2018).

Os projetos desenvolvidos com Arduino podem ser executados mesmo sem a necessidade de estar conectados a um computador, apesar de que também podem ser feitos comunicando-se com diferentes tipos de software (como Flash, *Processing* ou MaxMSP).

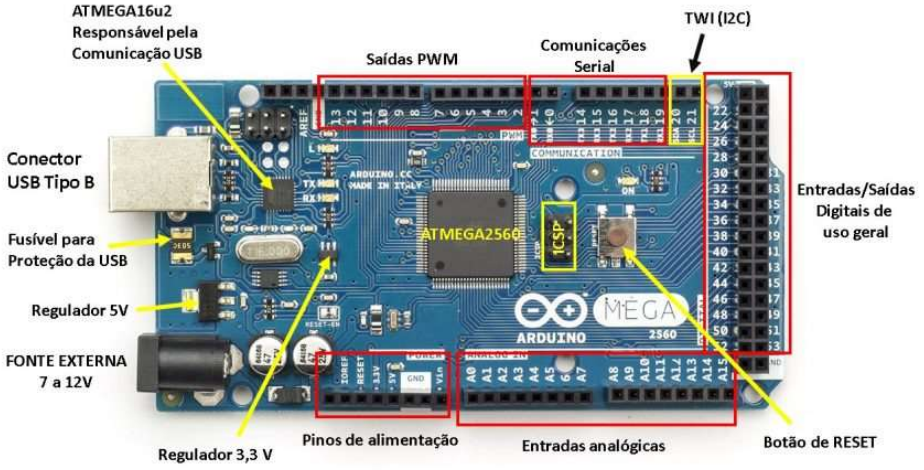
Além da quantidade de pinos, o Arduino MEGA conta com maior quantidade de memória que Arduino UNO, sendo uma ótima opção para projetos que necessitem de muitos pinos de entradas e saídas além de memória de programa com maior capacidade. A figura 10 ilustra um Arduino com resumo de recursos da Arduino MEGA 2560.

> O Arduino Mega é uma placa com microcontrolador ATMEGA2560. Ele possui 54 pinos de entradas/saídas digitais, 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada de alimentação e um botão de reset. Ele contém tudo o que é necessário para dar suporte ao microcontrolador, basta conectar a um computador com um cabo USB ou a uma fonte de alimentação e já está pronto para começar. Alimentação: A placa pode operar com alimentação externa entre 6 e 20 volts. No entanto, se menos de 7 volts forem fornecidos, o pino de 5V pode fornecer menos de 5 volts e a placa pode ficar instável. Com mais de 12V o regulador de voltagem pode superaquecer e danificar a placa, ou seja, a faixa recomendável é de 7 a 12 volts. Memória: O ATmega2560 tem 256 KB de memória flash para armazenamento de código (dos quais 8 KB é usado para o bootloader),8 KB de SRAM e 4 KB de EEPROM (que pode ser lido e escrito com a biblioteca EEPROM) (SOUZA, 2017).

De acordo com Sousa (2017)

> Entrada e Saída: Cada um dos 54 pinos digitais do Mega pode ser usado como entrada ou saída, usando as funções de pinMode(),digitalWrite(), e digitalRead(). Eles operam a 5 volts. Cada pino pode fornecer ou receber um máximo de 40 mA e possui um resistor interno (desconectado por default) de 20-50KΩ. Em adição alguns pinos possuem funções especializadas. Comunicação: O Arduino Mega possui várias facilidades para se comunicar com um computador, com outro Arduino ou outro microcontrolador. Os LEDs RX e RT piscarão enquanto dados estiverem sendo transmitidos pelo chip FTDI e pela conexão USB ao computador (mas não para comunicação serial nos pinos 0 e 1).

Figura 10 – Resumo de recursos da Arduino MEGA 2560.



Fonte: SOUZA (2017).

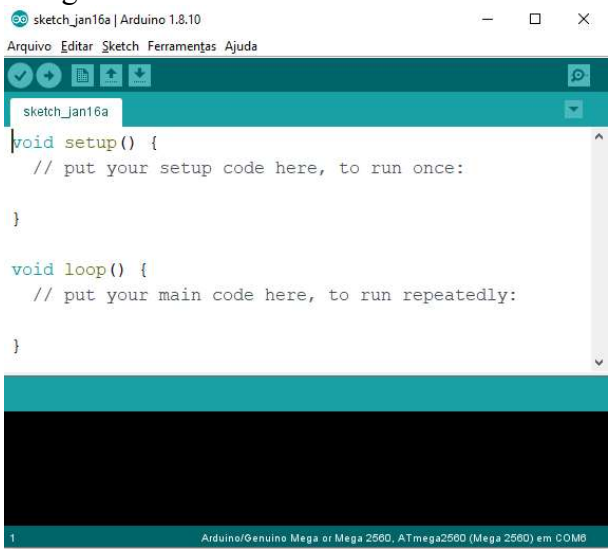
Quando se há uma necessidade de uma posterior expansão do projeto, quando se vislumbra um incremento ao código do programa e dessa forma uma maior aplicabilidade do que será executado, a placa Arduino Mega 2560 é sempre uma boa alternativa devido ao aumento do número de pinos e da quantidade de memória FLASH. Com suas entradas analógicas e saídas PWM a mais, possui uma gama maior de possibilidades para a inserção de recursos extras, sendo atuadores ou sensores (MCROBERTS, 2011).

O Arduino NANO funciona exatamente como um arduino comum, UNO, porém com o tamanho reduzido, podendo facilmente caber em qualquer espaço. Pode ser colocado como um CI (Circuito Integrado) em uma placa de circuito impresso ideal para protótipos que precisam de uma interface mais profissional.

3.10.1 Interface de Programação – Software IDE do Arduino

Utilizamos a IDE do Arduino para realizar ações através dos atuadores e “sentidas” através dos sensores. O código é escrito no ambiente computacional mostrado na figura 11, chamado de sketch. A conexão é feita diretamente do computador com o Arduino via USB, permitindo o *upload* de programas para o microcontrolador.

Figura 11 – IDE do Arduino versão 1.8.12



Fonte: Arduino (2020).

Na interface é possível observar que o ambiente em branco é onde a programação deverá ser implementada. A sintaxe da linguagem Arduino foi desenvolvida por Hernando Barragan; é basicamente C/C++ e possui funções simples e específicas para uso das portas do Arduino. São necessárias duas funções elementares para seu funcionamento: *setup ()* e *loop ()*.

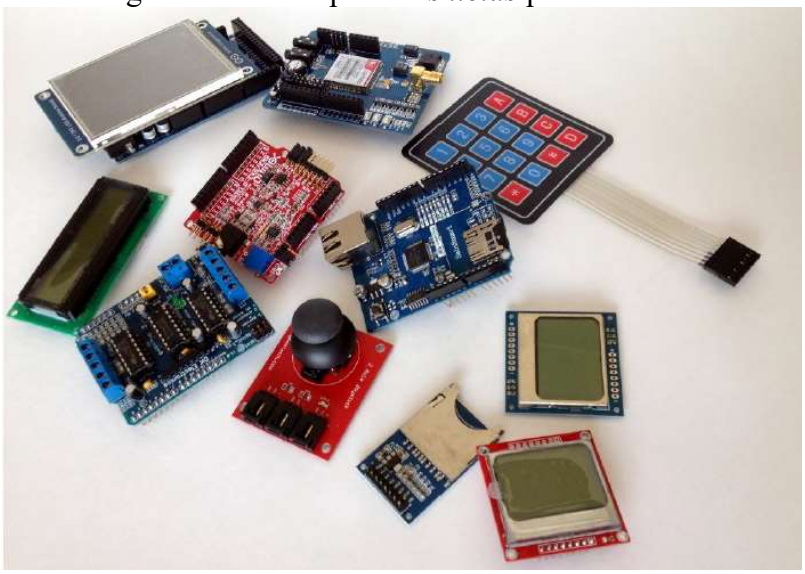
O Arduino utiliza uma linguagem de programação simples, adaptada da linguagem *Processing*, que é chamada de *Wiring*. O que torna essa linguagem simples é o fato de quando pressionado o botão de *upload*, o código escrito é traduzido para a linguagem C (normalmente de difícil compreensão por quem está iniciando) (BANZI, 2010).

> O Wiring é um sistema de código aberto (hardware e software) criado por Hernando Barragán no Interaction Design Institute Ivrea, na Itália. Essa plataforma serviu de base para criação da plataforma Arduino e ficou conhecida como a Linguagem Wiring. Essa linguagem foi desenvolvida com base no Processing e trata-se de uma abstração de hardware desenvolvida em C/C++ para programação de microcontroladores. (CURVELLO *et al*, 2017).

3.10.2 Shields

Uma placa Arduino possui funções restritas e específicas, necessitando, dessa forma, do acoplamento de outros circuitos que alavancam o potencial do microcontrolador. Tais circuitos são chamados de s*hields*. O Arduino pode se conectar à internet, ser um receptor GPS, controlar motores, simular funções de um telefone ou até MP3 *Player*, portanto é um importante componente da placa principal. Além de serem de distribuição *open source* igual a plataforma Arduino. A figura 12 mostra os diferentes modelos existentes no mercado.

Figura 12 – Exemplos de *shields* para Arduino.



Fonte: LEMOS (2013).

3.11 Comunicação SPI

As comunicações síncronas são muito importantes na transmissão de dados entre dois dispositivos. SPI significa *Serial Peripheral Interface* (Interface Periférica Serial) e trata-se de uma comunicação serial síncrona para pequenas distâncias, isto é, ela utiliza uma linha para sincronizar os dados entre o dispositivo que envia e o dispositivo que o recebe. A SPI não define um protocolo para interpretar os dados enviados/recebidos (RODRIGUES, 2016).

A SPI é uma comunicação do tipo mestre-escravo (*máster-slave*), em que pode existir apenas um mestre (quem controla a comunicação) e que podem existir inúmeros escravos (quem é controlado).

A comunicação SPI, além do mestre, tem outros chamados de escravos. Como o Arduino foi o mestre, os outros periféricos foram os escravos (MENDONÇA, 2020). Esta comunicação dá por meio de um barramento e constitui-se de no mínimo 4 conexões diferentes:

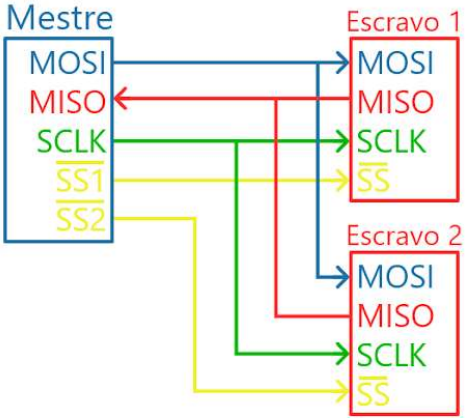
- SI ou MOSI (*Master OUT Slave IN*) – Entrada Serial. Linha de recepção de dados. Dispositivo mestre envia os dados e o dispositivo escravo os recebe - Master para *Slave*;
- SO ou MISO (*Master IN Slave OUT*) – Saída Serial. Linha de transmissão de dados. Dispositivo mestre lê os dados que o dispositivo escravo envia - *Slave* para Master;
- SCK (*Serial Clock*) - Clock de sincronização - transmissão dados entre o Master e *Slave*; Utilizada para sincronização entre o dispositivo mestre e o dispositivo escravo. O sinal deve ser simétrico (*duty* de 50%) e a informação de saída do sinal deve estar disponível

no mínimo por 30ns antes da borda de subida do clock e lida até 30ns antes da borda de descida do clock. A validação do dado dá-se normalmente na borda de subida.

- CS (ChipSelect) ou SS (*SlaveSelect*) - Seleciona qual *Slave* receberá os dados. A seleção de dispositivo é usada para selecionar ou habilitar o dispositivo com o qual se deseja comunicar e também para encerrar a execução dos comandos transmitidos pelo dispositivo mestre (SOUSA, 2020).

A figura 13 apresenta as ligações entre um mestre e dois escravos, mostrando a entrada serial MOSI, saída serial MISO e clock de sincronização SCLK comuns do mestre para os escravos e uma saída diferente para cada escravo.

Figura 13 – Ligações entre um mestre e dois escravos.



Fonte: (SOUSA, 2020)

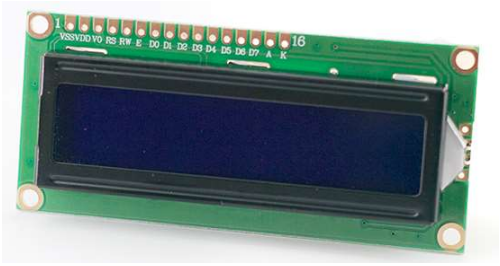
3.12 Display LCD

Os displays LCD são muito úteis para quem pretende usar um microcontrolador para desenvolver uma aplicação. Eles permitem uma interface visual entre homem e máquina, barata e simples de usar (MURTA, 2018).

O display LCD 16x2 (Figura 14), apresenta 6 colunas por 2 linhas, *backlight* azul e escrita branca, é um display muito comum com controlador HD44780 e se adapta aos mais diversos projetos, podendo ser usado com vários modelos de placas e microcontroladores como Arduino, Raspberry Pi, PIC, etc (THOMSEN, 2011).

O display LCD, de acordo com a biblioteca usada na IDE do Arduino, o seguinte padrão de pinos: LiquidCrystallcd(<pino RS>, <pino *enable*>, <pino D4>, <pino D5>, <pino D6>, <pino D7>).

Figura 14 – Display LCD 16x2.



Fonte: THOMSEN (2011).

O display LCD tem 16 colunas e 2 linhas, com *backlight* (luz de fundo) azul e letras na cor branca. Para conexão, foram 16 pinos, dos quais usamos 12 para uma conexão básica, já incluindo as conexões de alimentação (pinos 1 e 2), *backlight* (pinos 15 e 16) e contraste (pino 3). Todas as conexões são mostradas na tabela 1.

Tabela 1 – Conexões LCD 16x2 – HD44780.

Pino LCD	Função	Ligação
1	Vss	GND
2	Vdd	Vcc 5V
3	V0	Pino central potenciômetro
4	RS	Pino 12 Arduino
5	RW	GND
6	E	Pino 11 Arduino
7	D0	Não conectado
8	D1	Não conectado
9	D2	Não conectado
10	D3	Não conectado
11	D4	Pino 5 Arduino
12	D5	Pino 4 Arduino
13	D6	Pino 3 Arduino
14	D7	Pino 2 Arduino
15	A	Vcc 5V
16	K	GND

Fonte: THOMSEN (2011).

3.13 Sensores de Contato – Chave Fim de Curso

Estes tipos de dispositivos, muito importantes para gerarem pulsos elétricos e assim darem seguimentos a partes de código no Arduino, são eletromecânicos, isto é, seu princípio de funcionamento é baseado na detecção de uma mudança do contato físico direto com um obstáculo ou objeto que seja o alvo da permuta de estado. Os grandes representantes deste ramo são as chaves fim de curso e as chaves de segurança.

Essas chaves são basicamente constituídas por uma alavanca ou haste, com ou sem roldanas na extremidade, que transmite o movimento aos contatos que se abrem ou se fecham de acordo com a sua função. Pode ser na parte de controle, sinalizando os pontos de início ou de parada de um determinado processo, e de segurança, desligando equipamentos quando há abertura de porta ou equipamento e alarme (FRANCHI e CAMARGO, 2008).

Os sensores são responsáveis por informar e receber qualquer mudança de estado, como alteração na temperatura do ambiente ou acionamento de uma sirene pelo sistema de segurança.

As chaves fim de curso detém desse nome por conta de poderem determinar a presença ou a ausência, o posicionamento, normalmente indicando o final de um percurso percorrido por um objeto, indicando ainda a passagem de um obstáculo ou final de um trajeto. São mostrados na figura 15 vários tipos de chaves de curso, demostrando seu amplo uso em diversos campos da eletroeletrônica.

Necessitam de um contato físico para, assim, fechar o contato elétrico e gerar pulso elétrica para alguma parte do circuito que estava isolada. Essas chaves são muito utilizadas em portas automáticas, como de elevadores, e no caso deste trabalho, será utilizada para gerar pulso elétrico quando o motor de passo, juntamente com o aerógrafo, chegar ao final do percurso dentro da câmara de aquecimento.

Por conta disto, é necessário considerar fatores como o espaço disponível para instalação, o grau de proteção, tipo de atuador, tipo de contato NA ou NF, entre outros.

Figura 15 – Chaves Fim de Curso.

Fonte: TURK (2020).

3.14 Motor de Passo

Motores de passo (Figura 16) são dispositivos eletromecânicos que tem a capacidade de converter impulsos elétricos em movimentos discretos mecânicos (WEIHUA, 2007). São motores que, como a grande maioria dos motores elétricos, tem seu funcionamento baseado na interação entre campos eletromagnéticos (QUEIROZ, 2002).

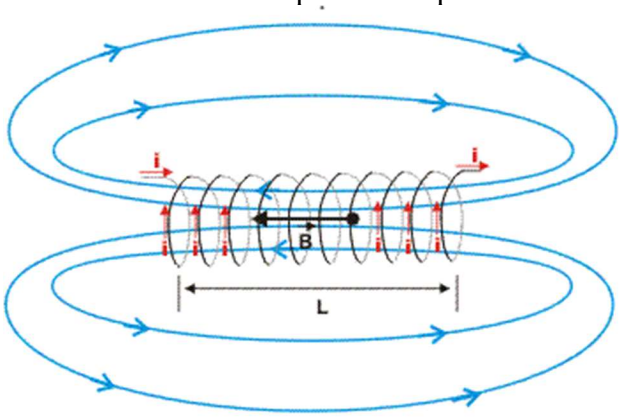
Os motores de passo caracterizam-se por ter a movimentação relacionada a deslocamentos angulares, denominados passos. São muito utilizados na eletrônica por possuírem muita força, muita potência, um grande torque e podem ser facilmente controlados (AGNIHOTRI, 2011).

Seu princípio básico é baseado em uma corrente elétrica percorrendo um solenoide, como mostrado na figura 17, e, com isso, gera-se um campo magnético uniforme em seu interior. Inverte-se o sentido do campo magnético invertendo o sentido da corrente elétrica (CONDIT, 2004).

Figura 16 – Tipos de Motor de Passo.

Fonte: Usina Info (2019).

Figura 17 – Solenoide sendo percorrido por corrente elétrica.



Fonte: Alfa Connection (2020).

Com os motores de passo há a possibilidade de se controlar sua velocidade, direção e ângulo, pode-se girá-los num ângulo determinado com extrema precisão, que varia de acordo com o tipo e que energia necessária ao seu projeto. Eles podem ser encontrados em dois principais tipos diferentes: magnético permanente e relutância variável. Há também a combinação entre eles, o motor de passo híbrido, que são os mais utilizados em aplicações industriais (KOSOW, 1982).

Uma característica marcante deste tipo de motor é a ausência de escovas (FALCONE, 1979). Segundo Agnihotri (2011), os comutadores e escovas dos motores convencionais são os componentes que apresentam a maior parte das falhas e ainda podem criar arcos voltaicos que são indesejáveis e perigosos em alguns ambientes. Dessa forma já temos uma minimização destes riscos quando são utilizados motores de passo. Não há a análise do circuito em malha fechada, não sendo necessário uma atenção ao circuito de realimentação, isto é, nenhuma informação de feedback sobre a posição é imprescindível.

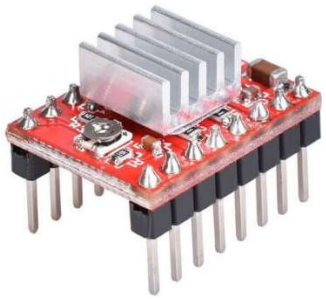
No projeto do forno, sua eficácia está mais na exatidão de como é feita a movimentação do objeto que realiza a técnica do spray pirólise. Para a realização da operação de velocidade e posicionamento mais precisos, necessitam de um controle mais complexo. Assim, para controlar os motores de passos, pode-se usar circuitos digitais para determinar a sequência de acionamento das bobinas do motor e a partir disso, determinar a velocidade de rotação, direção e posicionamento.

A rotação desses motores tem várias relações diretas com os pulsos de entrada aplicados. A sequência dos impulsos elétricos aplicados está diretamente relacionada à direção de rotação do eixo do motor; a velocidade de rotação do motor está diretamente relacionada à frequência dos impulsos de entrada. O comprimento de rotação está relacionado ao número de impulsos da entrada. É fundamental saber que para se controlar um motor de passo é de extrema importância a aplicação da tensão a cada uma das bobinas em uma específica sequência. (PATSKO, 2006).

3.15 Drive A4988

O drive de motor de passo A4988 (Figura 18), auxilia o controle dos motores utilizando dois pinos para controle e com a possibilidade de movimentar os motores por meio de micropassos, que é o procedimento básico para sua aplicação que exige um posicionamento com exatidão do motor.

Figura 18 – Drive de motor A4988.



Fonte: Acelera 3D (2020).

A alimentação do drive varia entre 3 V e 5,5 V e para o motor uma alimentação externa de 8 V a 35 V. A regulagem da corrente de saída do dispositivo é feita por meio de um potenciômetro e a disposição dos pinos é mostrado na figura 19.

Figura 19 – Pinagem A4988.

Fonte: Arduino & Tecnologia (2020).

O acionamento dos motores de passo pelos pinos é feito por meio dos modos apresentados na tabela 2, com os modos *full-step, half-step, quarterstep* (1/4), eightstep (1/8) e *sixteenthstep* (1/16). Os pinos configuráveis são MS1, MS2 e MS3. A utilização do modo micropasso reduz o torque do motor, em média, 30% em comparação com o modo de passo completo (*full-step*) (CONSTANDINOU, 2003).

Tabela 2 – Modo de operação do Motor de Passo.

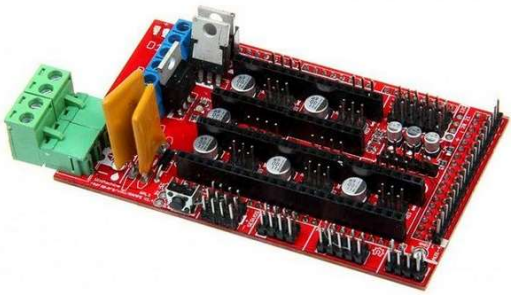
Pinos			
MS1	**MS2**	**MS3**	**Resolução micropassos**
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step (1/4)
High	High	Low	Eighth step (1/8)
High	High	High	Sixteenth step (1/16)

Fonte: Arduino e Cia (2020).

3.16 RAMPS

Ilustrado na figura 20, RAMPS é uma placa eletrônica, caracterizada como *shield*, contendo a eletrônica básica utilizada em máquinas de impressão 3D. O RAMPS é plugado no Arduino Mega, capacitando-o para mais funções, expandindo as operações do microcontrolador. Projetado para encaixar todos os componentes eletrônicos necessários para um RepRap em um pequeno pacote por um baixo custo. O RAMPS faz interface com o Arduino Mega com a poderosa plataforma MEGA do Arduino e tem muito espaço para expansão. Várias placas de expansão Arduino podem ser adicionadas ao sistema, desde que a placa RAMPS principal seja mantida no topo da pilha.

Figura 20 – Ramps - RepRap Arduino Mega.



Fonte: Auto Core Robótica (2020).

3.17 Sensor de Temperatura

A temperatura é uma medida da energia cinética medida em uma amostra de material expressa em unidades de graus em uma escala padrão. Pode-se medir temperatura de muitas formas, as quais variam em preço e precisão do equipamento. Os modelos mais comuns de sensores são termopares, resistências metálicas (RTDs), e termistores.

3.17.1 Termopar

Termopares, como mostrado na figura 21, são sensores de temperatura simples, robustos e de baixo custo, sendo amplamente utilizados nos mais variados processos de medição de temperatura. Um termopar é constituído de dois metais distintos que unidos por sua extremidade formam um circuito fechado.

Figura 21 – Termopar tipo K.

Fonte: Athos Electronics (2020).

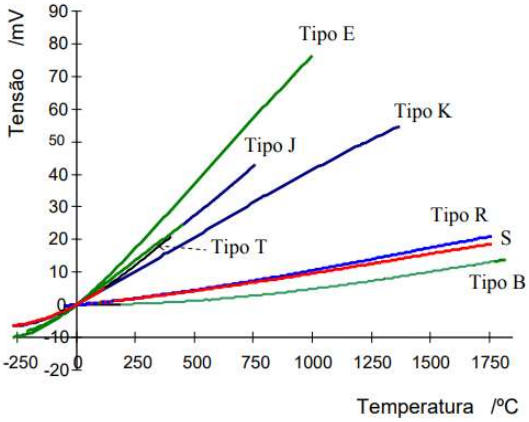
O termopar gera uma Força Eletromotriz (FEM) que, quando conectada a um Instrumento de Leitura, consegue ler a temperatura do processo destes Termopares. Este dispositivo é formado por dois materiais metálicos diferentes que, quando uma de suas extremidades estão unidas, forma-se uma conexão que irá gerar uma pequena tensão devido à variação de temperatura.

Segundo Justi (2009), seleciona-se o melhor tipo de termopar para cada aplicação baseando-se na temperatura de processo, nas características ambientais, precisão e custo. Os tipos mais comuns de termopar são:

- Tipo K: Tem baixo custo e cobre temperaturas entre -200 e 1200ºC.
- Tipo E: Adequado para temperaturas abaixo de 0ºC.
- Tipo J: Sua faixa de temperatura está entre -40 e 750ºC.
- Tipo N: Adequado para medições de temperaturas elevadas.

O termopar do tipo K é formado por fios de Alumel, como termoelemento negativo, e Chromel, como termoelemento positivo. É o termopar mais adequado para medições contínuas entre -200ºC até 1200°C. É recomendado para uso contínuo em atmosferas oxidantes ou completamente inertes. O termopar de tipo K foi selecionado para o projeto por atender à faixa de temperatura desejada. Na figura 22 é apresentado a variação da temperatura de acordo com a tensão de saída e com o tipo.

Figura 22 – Tensões de saída, usando uma junção de referência a T = 0 ºC.



Fonte: SILVA (2018)

3.18 MAX6675

O circuito integrado MAX6675, como mostrado na figura 23, possui um conversor AD de 12 bits de resolução, um sensor para compensação de junta fria, um controlador digital e a saída dos dados de temperatura já convertidos via protocolo SPI (*Serial Peripheral Interface*) para conexão a qualquer tipo de microcontrolador (AVELAR, 2019).

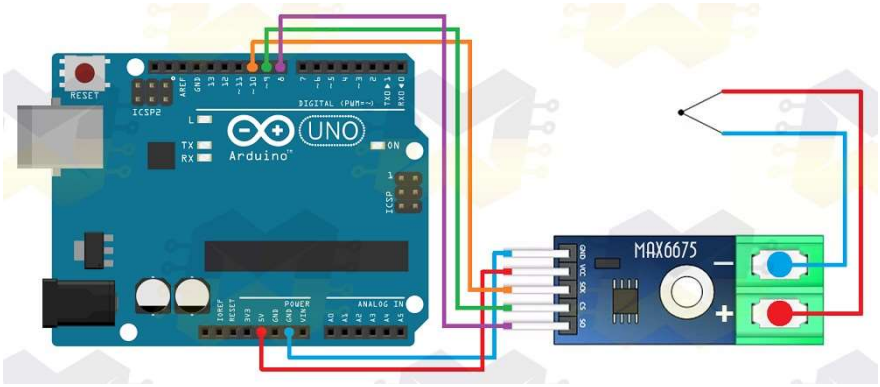
Figura 23 – Módulo MAX6675 para termopar.

Fonte: MADEIRA (2018).

Os termopares são versáteis e para cada variação na temperatura ocorrer uma pequena variação na tensão existente nos terminais do termopar, dispositivos, como o circuito integrado MAX6675, fazem o papel de intermediador, realizando o contato entre um Arduino e um termopar de maneira fácil.

O Módulo MAX6675 Termopar Tipo K – 0º a 800ºC é um dispositivo eletrônico que tem como finalidade fazer a medição de temperatura. A sonda que faz parte do módulo é toda revestida em aço inoxidável e a ponta possui blindagem. Esta forma de construção do termopar permite que o mesmo possa ser utilizado em temperaturas mais altas. A figura 24 apresenta o esquema de ligação somente do MAX6675 com o Arduino e o termopar. (AVELAR, 2019).

Figura 24 – Esquema de ligação de um termopar e drive Max6675 no Arduino UNO.



Fonte: OLIVEIRA (2019).

3.19 Linguagem de Programação

3.19.1 Linguagem de programação C

Citado por Deitel (1994), o C foi projetado e implementado em 1972 por Dennis Ritchie, sendo originalmente projetada para programação de sistemas operacionais, mas é adequada a uma ampla variedade de aplicações; tem muitas instruções de controle e facilidades de estruturação de dados; tem um vasto conjunto de operadores que permitem uma expressividade alta; e não possui verificação de tipo, portanto é uma linguagem muito flexível e ao mesmo tempo insegura.

3.19.2 Linguagem de programação C++

C++ é uma linguagem de programação criada por Bjarne Stroustrup no início da década de 1980. Com base em C, Hubbard (2003) acrescenta que é atualmente uma das linguagens mais populares para programação orientada a objetos. Foi padronizada em 1998 pelo American National Standards Institute (ANSI) e pela International Standards Organization (ISO). Possui o mecanismo classe/objeto, permite herança simples e herança múltipla e sobrecarga de operadores e funções.

A possibilidade de utilizar programação orientada a objetos permite ao programador projetar aplicações de um ponto de vista mais parecido com comunicação entre objetos que de uma sequência estruturada de código. Além disso, permite a reusabilidade de código de uma forma mais lógica e produtiva. A linguagem foi desenvolvida com o cuidado de prover atributos

Orientados a Objeto para a linguagem C sem comprometer a eficiência. Pode-se praticamente compilar o mesmo código C++ em qualquer tipo de computador e sistema operacional sem fazer grandes mudanças. C++ é uma das mais usadas e portadas linguagens de programação.

Código escrito em C++ é muito menor em comparação com outras linguagens, desde o uso de caracteres especiais e preferidos antes de palavras chave, evitando esforço (PANOSSO, 2019). Um corpo de aplicação em C++ pode ser feita de vários arquivos de código que serão compilados separadamente e "linkados" juntos. Economizando tempo, pois não é necessário recompilar toda a aplicação quando se faz uma mudança simples, mas apenas aquele arquivo que a contem (PEREIRA, 1999).

3.19.3 Linguagem de programação Python

Python é uma linguagem de programação interpretada, orientada a objetos, de alto nível e com semântica dinâmica (BORGES, 2010). A simplicidade do Python reduz a manutenção de um programa, suporta módulos e pacotes, que encoraja a programação modularizada e reuso de códigos.

É uma das linguagens que mais tem crescido devido sua compatibilidade (roda na maioria dos sistemas operacionais) e capacidade de auxiliar outras linguagens. Programas como Dropbox, Reddit e Instagram são escritos em Python. Python também é a linguagem mais popular para análise de dados e conquistou a comunidade científica.

Python é uma linguagem de propósito geral. Muitas vezes precisamos lidar com tarefas laterais: buscar dados em um banco de dados, ler uma página na internet, exibir graficamente os resultados, criar planilhas etc. Python tem vários módulos prontos para realizar essas tarefas.

Por esse e outros motivos que essa linguagem ganhou grande popularidade na comunidade científica. Além disso, Python é extremamente legível e uma linguagem expressiva, ou seja, de fácil compreensão. As ciências, por outro lado, possuem raciocínio essencialmente complicado e seria um problema adicional para cientistas conhecerem, além de seu assunto de pesquisa, assuntos complexos de um programa de computador como alocação de memória, gerenciamento de recursos etc. Python faz isso automaticamente de maneira eficiente e possibilitando o cientista se concentrar no problema estudado (BORGES, 2010).

É possível integrar o Python a outras linguagens, como a Linguagem C e Fortran. Em termos gerais, a linguagem apresenta muitas similaridades com outras linguagens dinâmicas, como Perl e Ruby.

4 MATERIAIS E MÉTODOS

4.1 Definição da linguagem de programação

Foi realizado o desenvolvimento de um sistema versátil que possui uma plataforma eletrônica aberta para a criação de protótipos baseada em software e hardware livres, flexíveis e fáceis de manusear.

O sistema foi otimizado no microcontrolador Arduino Mega, juntamente com s*hields,* Ethernet Shield, caso haja a necessidade de comunicação dos sensores com a web. O Arduino Mega centraliza todas as regras do sistema e processa todos os comandos conforme os acionamentos. Também foi utilizado o Arduino Nano e Uno para os circuitos de detecção de temperatura com os termopares, além de display LCD, módulo para cartão de memória, cartão de memória SD e módulos MAX6675 exatamente por seu tamanho reduzido.

O projeto contemplou, na IDE (Ambiente de Desenvolvimento Integrado) própria do Arduino, a implementação de um código utilizando a linguagem *Wiring* baseada em C/C++. Foi utilizado um sistema de comunicação com display LCD e teclado para receber as solicitações dos usuários para inserir as informações de movimentação da pistola aerográfica e encaminhar para o Arduino Mega, que irá processar os pulsos elétricos e acionar os atuadores, que serão os motores de passo, de acordo com as necessidades dos usuários.

4.2 Métodos

Foram criados códigos em Linguagem *Wiring*, comandado por Arduino MEGA e Nano e drive A4988, e em linguagem C, C++ e Python para a automação do forno elétrico inserido em uma câmara de aquecimento criado no Laboratório em Filmes Finos em Energias Renováveis – LAFFER. O código comandou 3 motores de passo:

- Motor 1 (M1) – Motor responsável por abrir e fechar a tampa do forno elétrico;
- Motor 2 (M2) – Motor responsável pelo movimento eixo x dentro e no alto da câmara, acoplado a um fuso trapezoidal de 8mm de diâmetro com 8mm de passo, uma guia linear de 10mm de diâmetro;
- Motor 3 (M3) – Motor responsável pela movimentação angular do aerógrafo, de modo que o spray pirólise deposite o líquido no substrato transparente.

4.3 Análise do Projeto

4.3.1 Análise Mecânica

Iniciou-se o processo de verificação da parte mecânica dentro da câmara (Figura 25). Essa etapa foi de imensa importância pois uniu-se a ela os procedimentos de como a técnica spray pirólise eram realizados manualmente.

Idealizou-se a fixação das engrenagens para a operação do aerógrafo na parte superior, realizando duas movimentações: uma seria seu movimento horizontal, sendo comandando por um motor de passo, e o outro movimento seria de rotação no próprio eixo, sendo comandado por outro motor de passo (Figura 26).

Figura 25 – Simulação de inclusão aerógrafo e motores de passo, fuso e guia cilíndrica.

Fonte: Autor.

Figura 26 – Simulação de inclusão motores de passo e aerógrafo.

Fonte: Autor.

Houve o estudo da área que foi abrangida pelo aerógrafo dentro da câmara de aquecimento, que se resume à área do forno elétrico, mostrado na figura 27, que mede 30 cm x 30 cm.

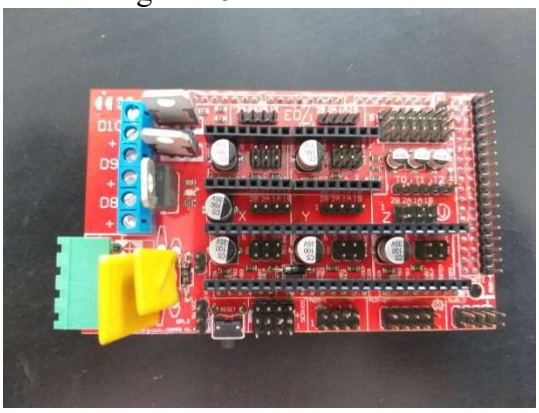
Figura 27 – Forno Elétrica e suas dimensões

Fonte: Autor.

4.3.2 Análise Elétrica

Foi realizado o acoplamento da RAMPS 1.4 (Figura 28) no microcontrolador Arduino.

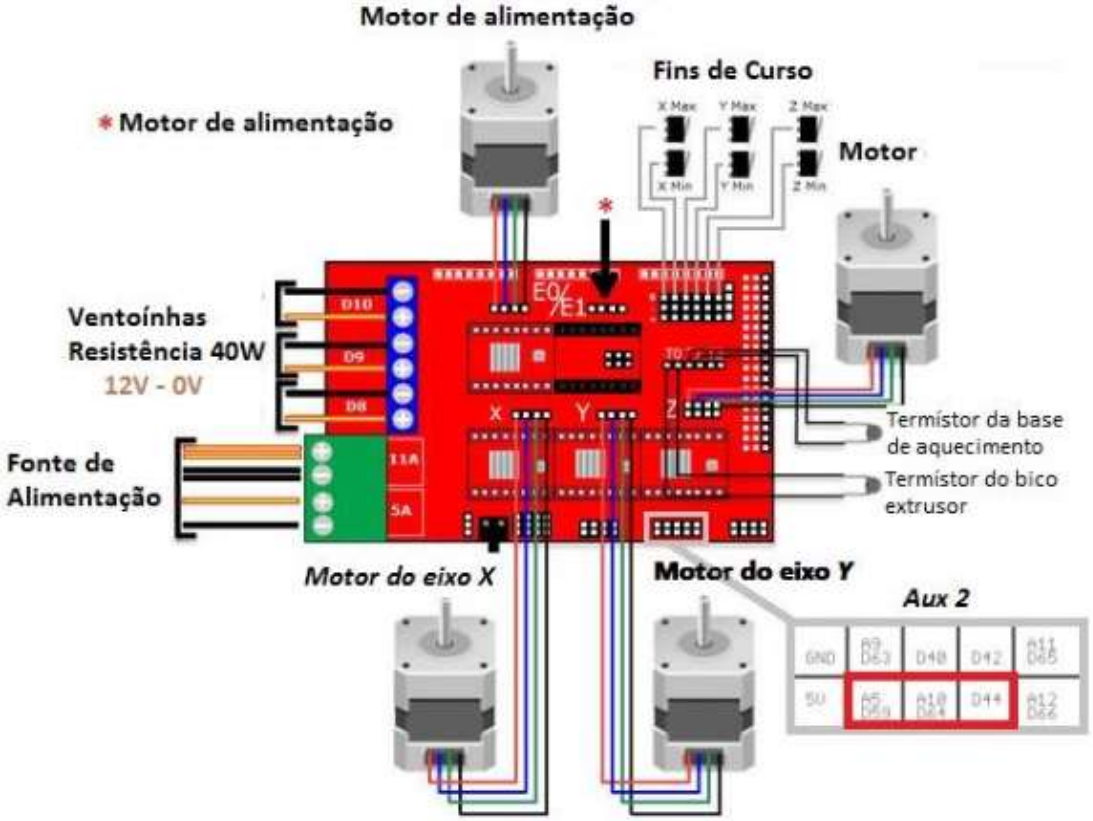
Figura 28 – RAMPS 1.4.



Fonte: Autor.

A identificação dos pinos da RAMPS 1.4 (Figura 29) possui 3 saídas controladas PWM de MOSFETS, controle mesa aquecida (*Heated bed*) com fusível de 11A, 3 circuitos de termistores, 5 Pololu *Stepper Driver sockets*, fusível de 5A para proteção da placa e componentes, pinos extras: PWM, digital, serial, SPI, I2C e analógico.

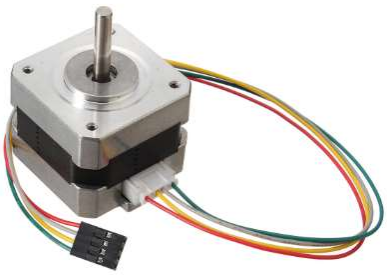
Figura 29 – Esquematização das ligações eletrônicas e identificação dos pinos auxiliares.



Fonte: REGADAS (2017).

Após isso, houve a ligação do cooler, do motor de passo (Figura 30). Então carregou-se o código implementado no IDE do Arduino para o microcontrolador.

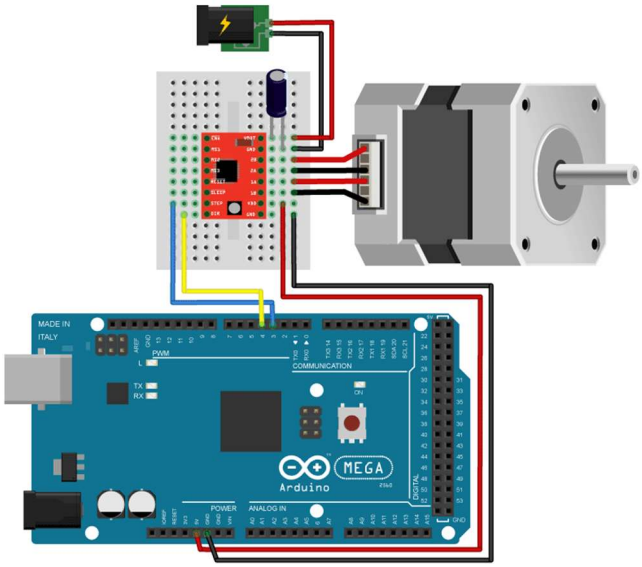
Figura 30 – Motores de Passo utilizados no projeto.



Fonte: Autor.

A figura 31 apresenta esquema de montagem que foi seguido e montado no protoboard com driver A4988, capacitor eletrolítico, fonte de alimentação, motor de passo e microcontrolador arduino.

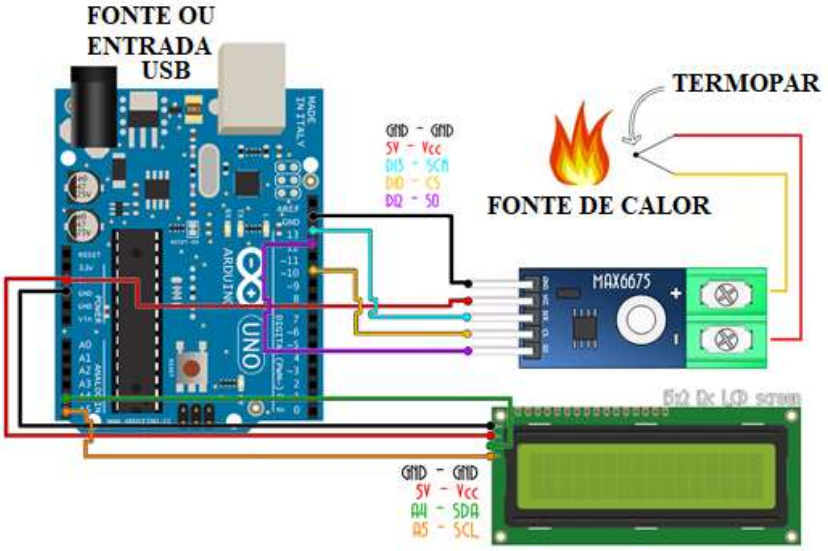
Figura 31 – Esquema de montagem no protoboard de fonte de alimentação, driver A4988, capacitor eletrolítico, motor de passo e Arduino.



Fonte: BAKKER(2019).

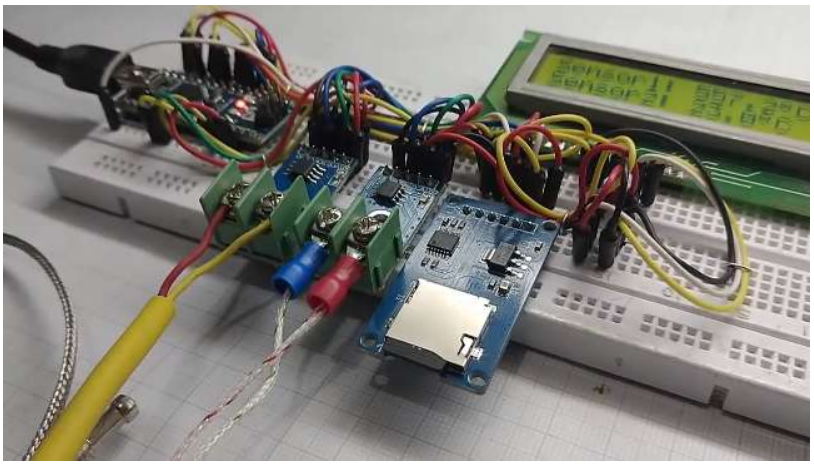
Outro circuito implementado é o que detecta a temperatura e dessa forma conseguir armazenar esses valores no cartão SD através do código implementado no IDE do Arduino. Na figura 32 é apresentado um esquema de ligação em que se simula a detecção de calor e, assim, o termopar detecta se há uma diferença de temperatura entre a extremidade unida e as extremidades livres, uma diferença de potencial surge e se o termopar estiver conectado ao dispositivo que tenha capacidade de interpretar o sinal, um valor de temperatura correspondente será apresentado. O módulo MAX6675 faz a medição dessa temperatura. O display LCD vai fazer a junção dos valores obtidos e enviados para o Arduino e dessa forma consegui mostrar visualmente no mesmo. Na figura 33 há a montagem física, com a implementação do código gerado nas linguagens que foram propostas no trabalho.

Figura 32 – Ligação Termopar, MAX6675, Arduino e Display LCD.



Fonte: Adaptado de Electronoobs (2019).

Figura 33 – Ligação física Termopar, MAX6675, Arduino e Display LCD.



Fonte: Adaptado de Avelar (2018)

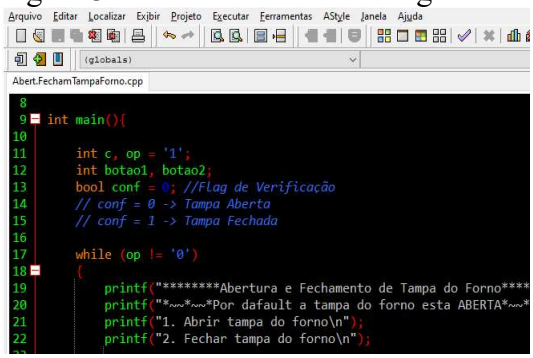
4.3.3 Análise Computacional

Foi feita a implementação dos códigos em Wiring, oriundos de C/C++, linguagem no IDE do Arduino. Primeiramente foi feito para a tampa do forno elétrico; segundo para o circuito de detecção das temperaturas e salvamento de informações; e finalmente para o circuito da movimentação do aerógrafo, mais complexo e denso.

Em seguida seguiu-se para a elaboração do código em linguagem C para o circuito da tampa do forno elétrico. Foi pensado estrategicamente nessa linguagem por haver possibilidades de os códigos serem usados, ou aprimorados, outros tipos de microcontroladores.

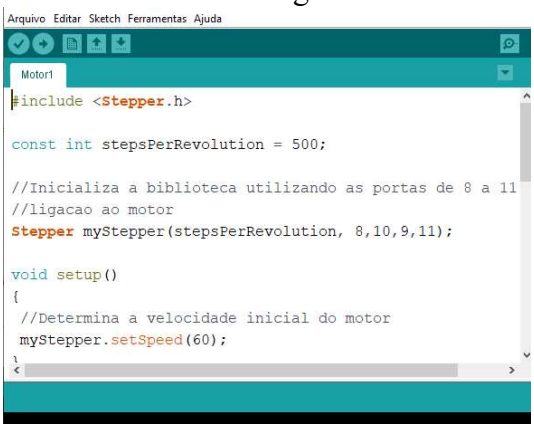
Pensando no aprofundamento dos usos dos circuitos embarcados para automação do forno, implementou-se códigos do circuito da tampa do forno em linguagem Python e C++, as mais utilizadas no meio acadêmico e científico. Nas figuras 34, 35, 36 e 37 são ilustradas as IDEs de cada linguagem usados para a implementação dos códigos.

Figura 34 – IDE DevC++ - Código em C.



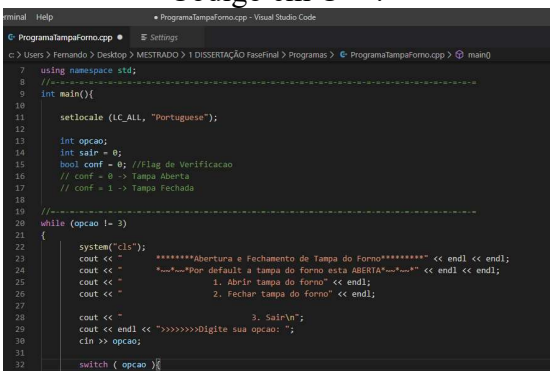
Fonte: Autor.

Figura 35 – IDE Arduino - Código em Wiring.



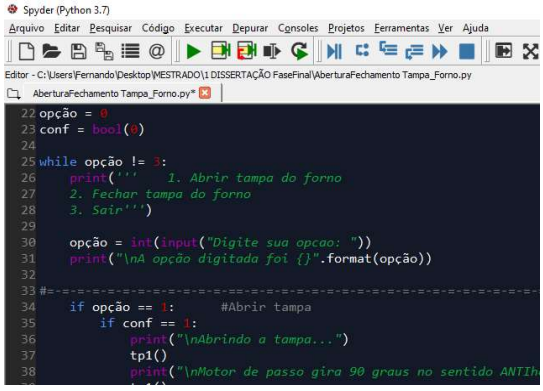
Fonte: Autor.

Figura 36 – IDE Visual Studio Code - Código em C++.



Fonte: Autor.

Figura 37 – IDE Spyder-Anaconda - Código em Python.



Fonte: Autor.

4.4 Materiais

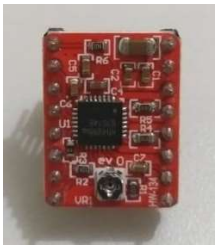
Para a parte mecânica foi utilizado os seguintes equipamentos e componentes:

- Câmara de Aquecimento;
- Forno Elétrico;
- 3 Motores de Passo Nema 17;
- Acoplador de eixo;
- Castanhas flangeadas em fuso;
- Castanha flangeada com mancais;

- Rolamento cilíndrico para guia linear;
- Arduino MEGA.

O drive A4988 é ilustrado na figura 38 com seus pinos, assim como na figura 39 temos a fonte de alimentação chaveada bivolt, com até 12 volts de saída, podendo usar o potenciômetro para indicar o valor de saída da tensão, de 20A e potência de 240W.

Figura 38 – Drive A4988.



Fonte: Autor.

Figura 39 – Fonte de alimentação chaveada 12V 20A 240W.

Fonte: Autor.

Para a ligação elétrica utilizou-se os seguintes componentes:

- Arduino MEGA;
- RAMPS 1.4;
- Fonte de alimentação chaveada bivolt 12V 20A 240W;
- Micro ventilador Cooler Ventoinha DC FAN 4010dc12hs;
- Drive A4988;
- Capacitor eletrolítico 100uF, 25V
- Cabos de ligação;
- Arduino UNO;
- Protoboard;
- Cabos de ligação.

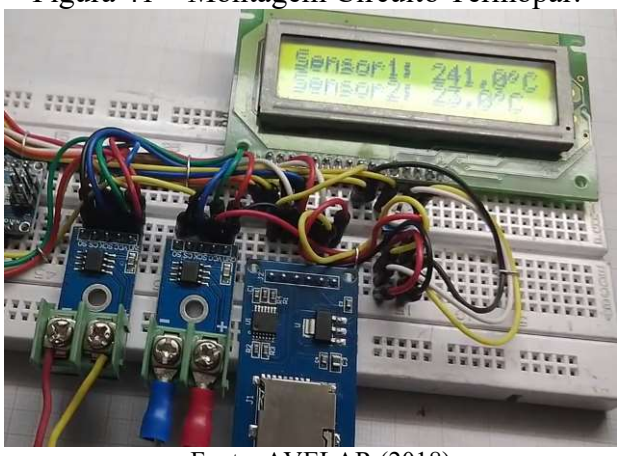
O esquema de montagem teve um note para a necessidade de um *shift-level* na entrada de dados do módulo do cartão pois o Arduino libera em sua saída um nível de 5V e este poderia danificar o cartão de memória que opera em 3,3V. A figura 40 ilustra o módulo do cartão SD.

Os materiais utilizados na montagem do circuito de detecção da temperatura com termopar foram: um Arduino Nano (Atmega 328); display LCD 16×2 (Controlador HD44780); potenciômetro de 4,7 KΩ; dois Termopares tipo K (0 a 600°C); módulo para cartão de memória com *Shift Level*; cartão de memória mini SD; dois módulos para leitura de termopar com o MAX6675 (0 a 1024°C); protoboard e finalmente cabos para ligação em protoboard. A figura 41 apresenta o circuito montado no protoboard utilizando dois sensores termopares.

Figura 40 – Módulo do cartão SD.

Fonte: AVELAR (2018)

Figura 41 – Montagem Circuito Termopar.



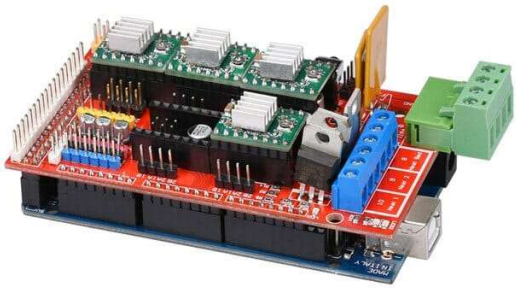
Fonte: AVELAR (2018).

5. RESULTADOS E DISCUSSÃO

5.1 Circuitos implementados

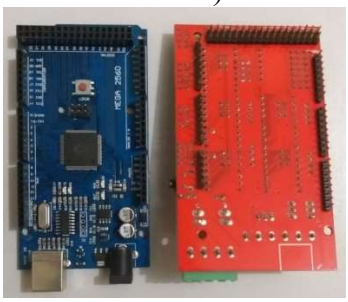
A montagem (acoplamento) das placas está mostrada na figura 42. São mostrados na figura 43 o arduino e a parte traseira da RAMPS e na figura 44 o arduino com a parte frontal da RAMPS.

Figura 42 – Acoplamento Arduino com RAMPS.



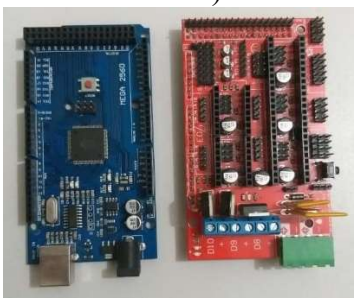
Fonte: Autor

Figura 43 – Arduino com RAMPS (parte traseira).



Fonte: Autor.

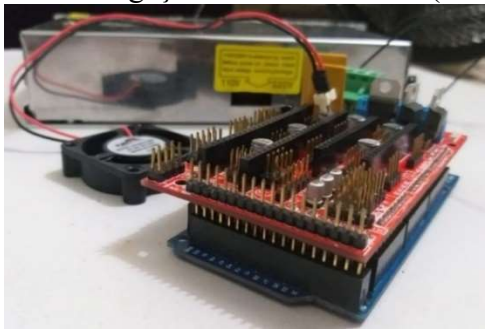
Figura 44 – Arduino com RAMPS (parte frontal).



Fonte: Autor.

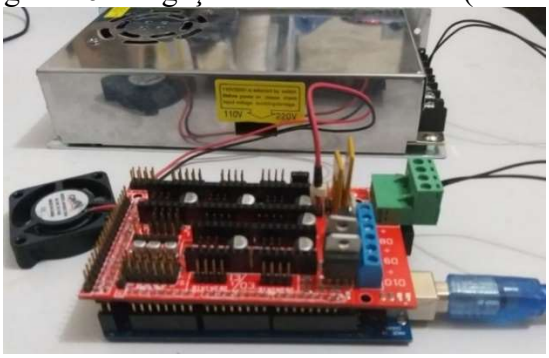
As ligações elétricas que foram realizadas no circuito do Arduino com RAMPS e a fonte de alimentação, mostradas nas figuras 45 e 46. A figura 47 apresenta o circuito implementado do um motor de passo com o arduino UNO, ligado a uma fonte de alimentação e sendo necessário a utilização de um capacitor eletrolítico.

Figura 45 – Ligação elétrica – vista 1 (frontal).



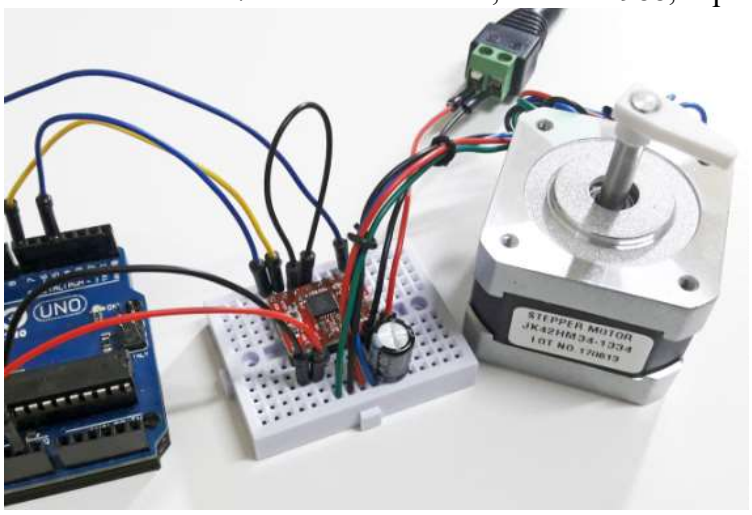
Fonte: Autor.

Figura 46 – Ligação elétrica – vista 2 (lateral).



Fonte: Autor.

Figura 47 – Circuito NEMA 17 com Arduino Uno, drive A4988, capacitor e fonte.



Fonte: Autor.

5.2 Construção e avaliação do SPA (Técnica de Spray Pirólise)

A deposição de filmes e soluções por spray pirólise é, de forma clássica, realizada manualmente para o controle dos principais parâmetros de deposição. Esta prática acarreta em uma série de imprecisões e incertezas, além de trazer grandes dificuldades para o processo de reprodução, uma vez que o método sendo manual traz uma dependência extrema de quem esteja operando e da sua perícia, cautela e cuidado.

Foi projetado um sistema eletromecânico que possibilita a movimentação do aerógrafo na coordenada X, sendo utilizada um motor de passo para essa finalidade, em tempo de deposição conferindo maior homogeneidade na distribuição do filme. O eixo Y também é controlado eletronicamente e foi representado como uma coordenada angular, pois ao invés do aerógrafo percorrer o eixo X, ele foi rotacional, permitindo maior praticidade no ajuste da distância entre aerógrafo e substrato. Toda parte eletromecânica foi microcontrolada pela plataforma Arduino.

5.2.1 Construção do sistema para movimentação do atomizador

Segundo Rodrigues (2008), a movimentação do atomizador em tempo de deposição confere maior homogeneidade, melhor distribuição de partícula e melhor condutividade iônica nos filmes depositados. O SPA utilizou dois motores de passo acionados por microcontrolador; um para promover movimento na coordenada cartesiana X e um outro na coordenada polar φ.

Um terceiro motor de passo foi usado para acionar a tampa do forno, sendo requisitado no início do processo, fechando o forno para o aquecimento dos substratos, durante o processo, abrindo e fechando a tampa do forno, e finalmente ao final do processo, em que ficará aberto para a retirada dos substratos da câmara de aquecimento.

O movimento em X é normalmente utilizado em tempo de deposição, perfazendo uma varredura ao longo da área depositada. Tanto o deslocamento no eixo X quanto no eixo φ são parâmetros configuráveis através da interface. Não há deslocamento no eixo Z.

5.2.2 Caracterização do gerador de spray

A solução usada foi uma solução aquosa ácida contendo cloreto de estanho II ($SnCl_2.2H_2O$), fluoreto de amônio (NH_4F), ácido clorídrico (HCl) e água. O processo manual foi realizado com um porta-substrato com inclinação de 25º, distância em média de 25 cm entre o aerógrafo e o substrato (LIMA, 2013). O processo automatizado procurou manter, em média, essas distâncias, pois houve a variação desse ângulo de 10° a 40° e distância de 10 cm a 20 cm.

A solução foi aspergida sobre o substrato em média por 3s em cada aplicação. Os substratos são lâminas para microscopia (vidro) aquecidas até uma temperatura T = 600ºC. Para produzir vidro condutor foram necessários alguns ciclos de operação.

Tais ciclos consistiram em retirar do forno o substrato a temperatura T = 600ºC, aspergir a solução sobre o substrato aquecido, recolocar o substrato no forno e esperar 2 min para reiniciar o processo. Além disso, em cada ciclo o número de aplicações variou. A tabela 3 mostra a quantidade de aplicações em cada ciclo, usando uma solução aquosa ácida (solução precursora) com volume de 50 ml.

- 1 aplicação equivale a uma varredura do forno pelo aerógrafo aspergindo a solução.
- Ciclo: Fecha tampa do forno → Aquece até 600 °C → Abre tampa → x aplicações, dependendo do ciclo.

Tabela 3 – Aplicações em cada ciclo.

Ciclos	Aplicações em cada ciclo
1°	1
2°– 3°	3
4° – 11°	6
12° – 16°	12

Fonte: Autor.

Todo o processo de automação pode ser conferido através da figura 48, em que detalhadamente mostra-se o passo a passo do processo e o início dos ciclos de aumento da temperatura até 600 °C e aspersão do líquido.

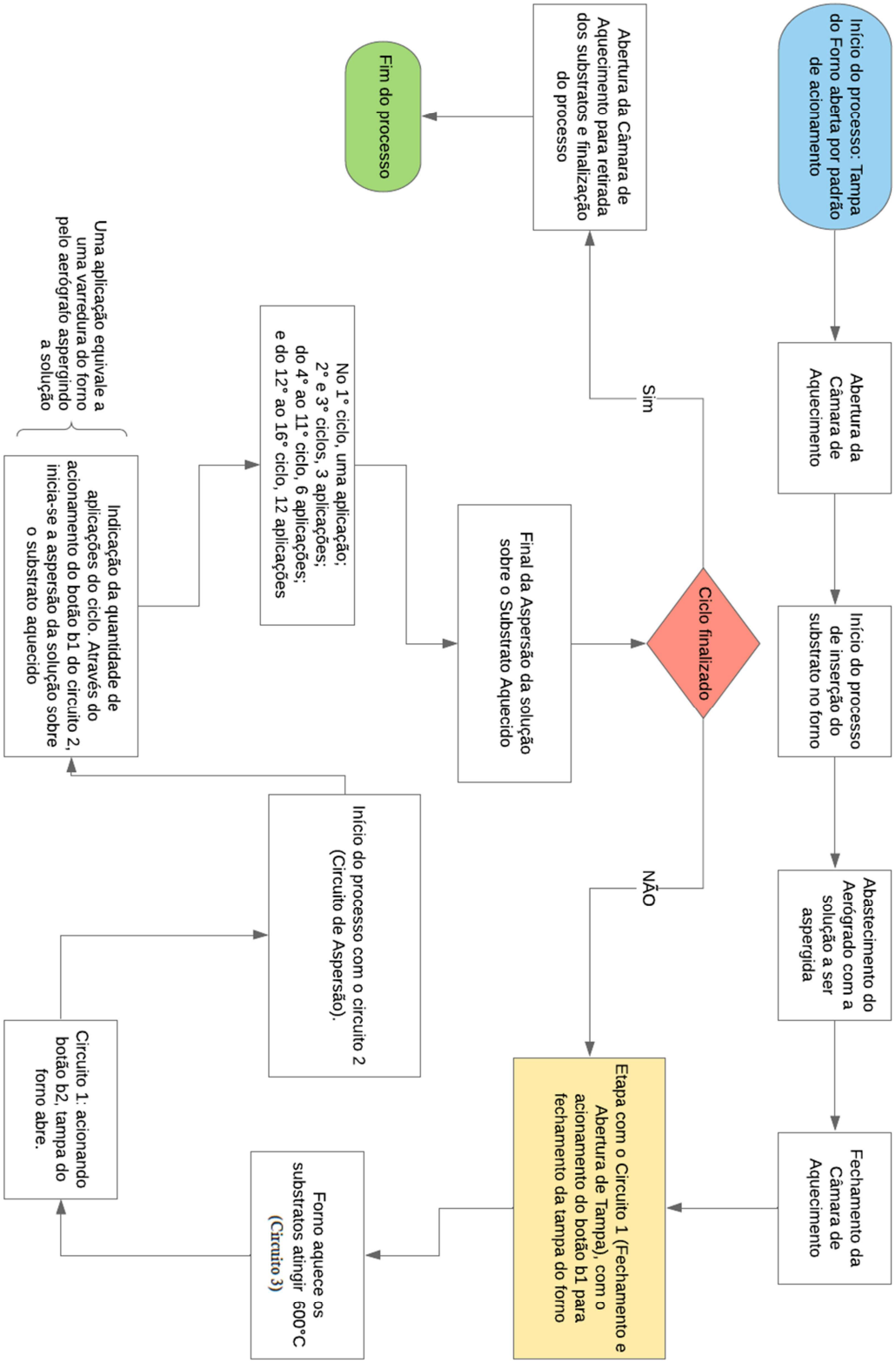
O uso de spray manual, isto é, um recipiente de spray em que a solução precursora é atomizada (spray), dispersa as gotículas por meio do movimento da mão humana, não ocorrendo de forma contínua. Apesar disso, o spray manual produziu filmes de resistência elétrica baixa. Apesar de fornecer resistência baixa, vidro não trincado durante o processo de deposição, maior razão líquido/ar, o vidro condutor apresentou regiões em forma de círculos espalhados sobre a superfície.

O uso do gerador de spray usando o aerógrafo de sucção e mistura interna produziu filmes com baixa resistência e superfícies relativamente homogêneas, mostrado em trabalhos anteriores (LIMA, 2013; MAIA Jr, 2015). O termo sucção é devido ao fato do recipiente no qual a solução precursora é colocada ficar na parte de baixo do aerógrafo e ocorre sucção da solução devido à passagem do ar por dentro do aerógrafo. Nas pistolas por gravidade o recipiente fica na parte superior. O termo mistura interna é devido ao fato da solução entrar em contato com o ar ainda dentro do aerógrafo.

Vidros condutores com alta condutividade elétrica, isto é, baixa resistência de filme (Rs), poucos micrômetros de espessura, área superficial homogênea e sem absorção na região do visível do espectro eletromagnético são adequados para o uso como eletrodos transparentes em células fotoeletroquímicas. As células fotoeletroquímicas construídas a partir dos vidros condutores fabricados por spray pirólise com resistência cerâmica como fonte de aquecimento do substrato apresentam efeito fotovoltaico.

A técnica spray pirólise permitiu obter filmes tanto sobre substratos isolantes elétricos quanto condutores. Em teoria com o uso dessa técnica, é possível depositar óxido semicondutor na forma de filme sobre qualquer tipo de substrato.

Figura 48 – Fluxograma do Processo de deposição de solução no substrato.



Fonte: Autor.

5.3 Circuito Termopar

5.3.1 Esquema de Ligação

O circuito termopar é o de detecção de temperatura dentro da câmara de aquecimento e representa o circuito 3 na figura 48. O código possui partes importantes a serem comentadas. Trata-se, portanto, de duas leituras de temperatura usando os termopares tipo K com módulo MAX6675, implementado no IDE do Arduino versão 1.8.10.

A primeira parte do código, nas primeiras linhas, apresenta a declaração das bibliotecas como LiquidCrystal.h, que permite que uma placa Arduino controle os monitores LiquidCrystal (LCDs) com base no chipset Hitachi HD44780 (ou compatível), encontrado na maioria dos LCDs baseados em texto. A biblioteca MAX6675.h também foi incluída por conta do uso do módulo de mesmo nome.

Outra biblioteca declarada é biblioteca SPI.h, que permite a comunicação com dispositivos SPI, com o Arduino como dispositivo mestre.

Alguns periféricos foram apenas escravos, por exemplo, cartão SD, memória flash e alguns sensores. MOSI - pin 11 no Arduino, MISO - pin 12 no Arduino, CLK - pin 13 no Arduino e CS1 - pin 10, CS2 - pin11 no Arduino.

A biblioteca SD.h permitiu a leitura e gravação em cartões SD, por exemplo, no *Shield* Ethernet do Arduino. A biblioteca suporta sistemas de arquivos FAT16 e FAT32 (*File Allocation Table*) em cartões SD padrão e cartões SDHC (SD *High Capacity*). Como o diretório de trabalho foi sempre a raiz do cartão SD, um nome se refere ao mesmo arquivo, incluindo ou não uma barra (por exemplo, "/file.txt" é equivalente a "file.txt").

No campo seguinte houve a declaração de variáveis. Para o MAX6675, tivemos pinSO para MISO, pinSCLK para serial clock, pinCS1 e pinCS2 para o Chip Select. Do cartão de memória tivemos a variável ChipSelect e finalmente para o sensor1 e sensor2, tivemos as variáveis dos sensores termopares.

O controle do display LCD foi realizado através da biblioteca LiquidCrystal.h, já embutida na IDE do Arduino. Assim, obteve-se a seguinte declaração de variáveis: LiquidCrystallcd(12, 11, 5, 4, 3, 2). No *setup*, inicializou-se o display definindo o número de colunas e linhas com o comando lcd.begin(16,2) – 16 colunas e 2 linhas.

Tivemos, a partir da biblioteca do LCD, a criação de até 8 caracteres personalizados. Cada segmento de caractere consistiu em um quadrado de 5x8 pixels. Gerou-se um símbolo de graus (°) com uint8_t degree[8] = {140, 146, 146, 140, 128, 128, 128, 128}. Como uint8_t é o

mesmo que um byte, com designação de um tipo de número inteiro não assinado de 8 bits de comprimento, pode-se ainda substitui-lo pelo próprio byte com variável grau. Como mostrado no código: byte grau[8] ={B00001100, B00010010, B00010010, B00001100, B00000000, B00000000, B00000000, B00000000,}.

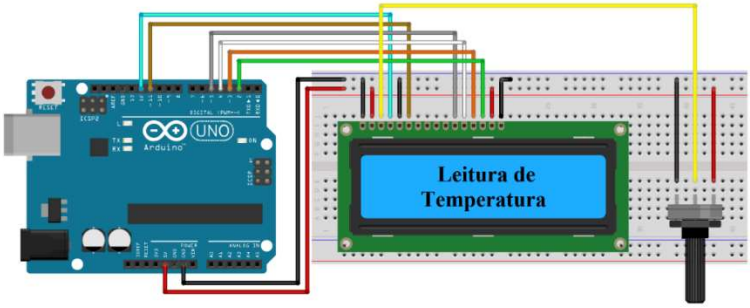
Três funções foram criadas e chamadas no início do código:

- **byte SPI_Leitura(void)** – Recebe vazio como parâmetro de entrada, usado para fazer a leitura do protocolo de dados serial SPI, usado pelos microcontroladores para se comunicar com um ou mais dispositivos periféricos rapidamente em curtas distâncias, podendo ser usados para comunicação entre dois microcontroladores;
- **double Leitura_Thermo(int ChipSelect) –** Recebe como parâmetro de entrada um inteiro do tipo ChipSelect, utilizada pra realizar as medições de temperatura advindas do MAX6675; e
- **void Armazenamento_SD()** – Não recebe nenhum parâmetro de entrada, implementando um SPI via Software para leitura dos dados no MAX6675.

Na função *setup*(), configurou-se a taxa de transferência em bits por segundo (*baud rate*) para transmissão serial, e ativa a porta serial em 9600 bps.

Nas funções LCD seguintes, definiu-se o número de colunas (16) e linhas (2) do LCD, criou-se um caractere personalizado para uso no LCD, posicionou-se o cursor na coluna 0 (zero) e linha 0 (zero), e 'printou' no mesmo a mensagem: LCD 16x2. Espera-se 1s (delay(1000)) e aparece outra mensagem: Leitura de Temperatura, como mostrado em seguida na figura 49.

Figura 49 – Exemplificação de Mensagem no Display LCD 16x2.



Fonte: Autor.

A próxima etapa do código garantiu se o cartão de memória está acoplado ao sistema e está pronto para ser utilizado, indicando 'Erro - Cartão Defeituoso' para trocar o cartão ou inseri-lo novamente e 'Cartão OK para uso" para o cartão com boas condições de uso.

5.3.2 Explanação do Código

A sessão *loop*() começou apagando o que tiver no display LCD para em seguida posicionar o cursor na coluna 0 (zero) e linha 0 (zero), aparecendo a palavra 'Sensor1:', a variável sensor1 mostrando a leitura do termopar através da função Leitura_Thermo(pinCS). Em seguida, aparece o símbolo de grau (°) e a letra 'C', indicando a temperatura Celsius. Depois ocorreu o mesmo com a variável sensor2. Finalizando a sessão *loop*() com a função Armazenamento_SD, em que ocorre o armazenamento da leitura realizada e aguarda 10 segundos para uma nova leitura e armazenamento.

Na função Armazenamento_SD(), declarou-se uma *string* para salvar os dados no SD, concatenando a palavra 'Sensor:' à temperatura mensurada pelo termopar, mais a *string* '°C', por exemplo: 'Sensor1: 550 °C' e 'Sensor2: 130 °C'.

A função SPI_Leitura(void) representou a leitura de um protocolo de dados serial SPI através de um laço *WHILE* para adquirir 8 bits de cada vez.

Na função Leitura_Thermo(intChipSelect), teve-se a passagem do parâmetro ChipSelect, CS do cartão de memória. Após isso, houve a declaração de uma variável de 16 bits, aux, com o tamanho ideal para o dado entregue pelo MAX. Na função *digitalWrite*(), se o pino for configurado como entrada (*INPUT*), a função *digitalWrite*() irá ativar (*HIGH*) ou desativar (*LOW*) o resistor interno de *pull-up* no pino de entrada. A leitura da parte alta, primeiros 8 bits de dados foi mostrada pela função SPI_Leitura().

Deslocou-se 8 posições para a esquerda e foi realizada a leitura da parte baixo e armazenou-a nos 8 primeiros bits através de uma lógica OR. Os três bits inferiores (0,1,2) foram bits de *status* descartados. Os bits restantes foram o número de contagens de 0,25 graus (°C).

5.4 Código de leitura de temperatura implementado.

```
#include <LiquidCrystal.h>
#include <max6675.h>
#include <SPI.h>
#include <SD.h>

const int pinSO = 8;
const int pinSCLK = 9;
const int pinCS = 10;
const int ChipSelect = 50;
double sensor1 = 0;
double sensor 2 = 0;
```

```
// Cria uma instância da biblioteca MAX6675.
//MAX6675 termoclanek(pinSCLK, pinCS, pinSO);

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

File dataFile; // Declara dataFile como arquivo para salvar no SD

//uint8_t degree[8]  = {140, 146, 146, 140, 128, 128, 128, 128};
 byte grau[8] ={ B00001100,
                 B00010010,
                 B00010010,
                 B00001100,
                 B00000000,
                 B00000000,
                 B00000000,
                 B00000000,};

// Declaração das função a serem utilizadas no decorrer do código.
byte SPI_Leitura(void);
double Leitura_Thermo(int ChipSelect);
void Armazenamento_SD();

void setup() {
Serial.begin(9600);
printf("Declaração das Variáveis e Iniciação do LCD\n");

  // Operação das portas
pinMode(pinCS1, OUTPUT);
pinMode(pinCS2, OUTPUT);
pinMode(pinSCLK, OUTPUT);
pinMode(pinSO, INPUT);
digitalWrite(pinCS1, HIGH);
digitalWrite(pinCS2, HIGH);

Printf(“Início Processo de Leitura de Temperatura”);
lcd.begin(16, 2);
lcd.createChar(0, grau);
lcd.setCursor(0, 0);
lcd.print("LCD 16x2");
delay(1000);
lcd.print("Processo de Leitura de Temperatura");
delay(1000);

Printf(“Conferência do cartão – Presente e em Bom estado.”);
if (!SD.begin(ChipSelect)) {
lcd.setCursor(0, 0);
lcd.print("Erro - Cartão Defeituoso");
return;
  }
```

```
else {
lcd.setCursor(0, 0);
lcd.print("Cartão OK para uso");
delay(500); }
}

void loop()
{
lcd.clear();        //Limpar o Display

printf("Cursor responsável pelo Sensor 1 - Próximo ao Forno Elétrico\n");
lcd.setCursor(0, 0);                                    //Linha 0, Coluna 0
lcd.print("Sensor1: ");
sensor = readThermo(pinCS1);
lcd.print(sensor1, 1);
lcd.write((byte)0);
lcd.print("C");

printf("Cursor responsável pelo Sensor 2 - Próximo à parte Superior da
Câmara\n");
lcd.setCursor(0, 1);                                    //Linha 1, Coluna 0
lcd.print("Sensor2: ");
sensor = readThermo(pinCS2);
lcd.print(sensor2, 1);
lcd.write((byte)0);
lcd.print("C");

Armazenamento_SD();
delay(10000);
}

// FUNÇÃO ARMAZENAMENTO DE DADOS

Void Armazenamento_SD() {
  // Formata os dados para serem salvos no Cartão SD a cada 1 segundo
printf("Impressão da Temperatura devido aos Sensores\n");
StringdataString = "";
dataString += "Sensor_1: ";
dataString += String(sensor1);       //Realização da Leitura do Sensor 1
dataString += " °C ";
dataString += "Sensor_2: ";
dataString += String(sensor2);       //Realização da Leitura do Sensor 2
dataString += " °C ";

// Abre o arquivo, escreve e o fecha novamente
  dataFile = SD.open("dadostemperatura.txt", FILE_WRITE);

  // Se o arquivo está presente, escreve no mesmo
  if (dataFile) {
    dataFile.println(dataString);
```

```
    dataFile.close();
  }
  // Se o arquivo não abrir, indicará no display
  else {
    lcd.setCursor(0, 0);
    lcd.print("Ocorreu alguma falha na Abertura do Arq.\n");
    lcd.setCursor(0, 1);
    lcd.print("O arquivo está no SD - OK.\n");
  }

}

// FUNÇÃO DE IMPLEMENTAÇÃO DE UM SPI

byte SPI_Leitura(void)
{
     int j = 7;
     byte r = 0;
printf("Implementação SPI via Software");

while(j >= 0)
{
digitalWrite(pinSCLK, LOW);  // Borda de descida do clock - Falling Edge
     _delay_ms(1);
      if (digitalRead(pinSO)) {    // Faz a leitura do pino de dados de saída do MAX
           r |= (1 << j);          // Armazena os bits lidos, 0 ou 1, na variável e desloca
                                   // de acordo com o índice para
   }                               // preencher todo o byte de dados
digitalWrite(pinSCLK, HIGH);     // Borda de subida do clock - Rising Edge
   _delay_ms(1);
  }j--;
  return r;                // Retorna o byte de leitura dos dados lidos do MAX
  }

// FUNÇÃO MEDIÇÃO DE TEMPERATURA - MAX6675

double Leitura_Thermo(intChipSelect)
{
uint16_t aux;       // Declara variável de 16 bits - Tamanho ideal para o dado
entregue pelo MAX
printf("Leitura dos dados de Temperatura do MAX6675");

digitalWrite(ChipSelect, LOW);
  _delay_ms(1);
aux = SPI_Leitura();       // Leitura da parte alta - Primeiros 8 bits de dados
aux<<= 8;                  // Desloca 8 posições para a esquerda
  aux |= SPI_Leitura();    // Faz leitura da parte baixo e armazena nos 8 primeiros
bits atravez de uma OR
digitalWrite(ChipSelect, HIGH);
```

```
Printf("Conferência do bit 2 para termopar acoplado ou não");
if (aux & 0x4) {
   return NAN;
}
aux>>= 3;   // Descarta os 3 primeiros bits, onde apenas o bit 2 carrega o status
e lê do bit 3 ao 14
Return aux * 0.25;  // Multiplica pela resolução do MAX
}
```

5.5 Circuito Tampa do Forno (Abertura e Fechamento da tampa)

O circuito 1, apresentado na figura 48, implementado foi composto por um motor de passo acoplado a peça que serviu de tampa do forno para o aquecimento dos substratos. Ele foi NA, isto é, Normalmente Aberto, pois a primeira ação foi de, após abrir a câmara de aquecimento, encontrar a tampa aberta para inserir os substratos (vidros) em cima do forno. Após essa ação, a seguinte etapa foi de fechar a câmara de aquecimento e iniciar o processo de deposição do substrato, clicando no botão 1 para fechamento da tampa e posteriormente, dando continuidade ao ciclo, botão 2 para realização da abertura do mesma. O código de implementação de abertura e fechamento da tampa, com uso de flags para garantir o intertravamento dos botões, é apresentado em seguida.

5.5.1 Código de Abertura e Fechamento da tampa do Forno Elétrico

```
/*-----------------------------------------------------------------------------------------------------------
    Abertura e Fechamento da Tampa do Forno com Intertravamento
-----------------------------------------------------------------------------------------------------------
  -- IDE do Arduino Versão 1.8.10
  -- Autor Fernando Oliveira
  -- email: fwsoliver@gmail.com
  -- Janeiro, 2020
-----------------------------------------------------------------------------------------------------------
*/

// ---------------------------------------------------------------------------------------------------------
//  ==== Bibliotecas Auxiliares ====
#include <A4988.h>              //Biblioteca Driver Motor de Passo A4988
#include <BasicStepperDriver.h>
#include <DRV8880.h>            //Biblioteca Driver DRV8880
#include <MultiDriver.h>
#include <SyncDriver.h>
#include <Stepper.h>            //Biblioteca para controle do motor de passo
// Biblioteca Motor de passo
#include "Stepper_28BYJ_48.h" // declaração biblioteca
```

```
// ---------------------------------------------------------------------------------------------------------
//  ==== Mapeamento de Hardware ====

#define in1 8        //Entrada 1 do ULN2003
#define in2 9        //Entrada 2 do ULN2003
#define in3 10       //Entrada 3 do ULN2003
#define in4 11       //Entrada 4 do ULN2003

//Definindo uma constante para indicar a quantidade de passos por revolução (volta)
//Motor de Passo KTC42HS40-1704XA-01F, ângulo de 1,8° por passo, e necessita de 200 passos
//para realizar uma volta completa. Do tipo Unipolar.

// ---------------------------------------------------------------------------------------------------------
//  ==== Constantes Auxiliares ====
const int stepsPerRevolution = 200;
//Para realizar 1/4 de volta (90 graus), seria necessário 50 passos.

int botao1_pin = 8; //pino do primeiro botão
int botao2_pin = 10; // pino do segundo botão

// ---------------------------------------------------------------------------------------------------------
//  ==== Declaração de Objetos ====

//Inicializando a biblioteca utilizando as portas 8,9,10,11 para ligação do motor
Stepper myStepper(stepsPerRevolution, in1, in3, in2, in4);
//Stepper myStepper(stepsPerRevolution, 8, 10, 9, 11);

/*
 * Flags são variáveis utilizadas como comando de controle de programa.
 * Isso significa que através dos valores das variáveis Flags controlamos a atuação ou sequência do programa.
*/
int ledPin1 = 13;
int ledPin2 = 12;
intbuttonState = 0;
int flag1 = 0; // varíavel de controle de programa
int flag2 = 0;

Stepper_28BYJ_48 stepper(7,6,5,4); // Declaração da instância do motor e os pinos de comando

// ---------------------------------------------------------------------------------------------------------
//Início da Sessão SETUP

void setup()
{
//declaração dos botões e ativação de seus resistores pullup internos
pinMode(botao1_pin,INPUT_PULLUP);
pinMode(botao2_pin,INPUT_PULLUP);
```

```
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
//pinMode(buttonPin, INPUT);
Serial.begin(9600); // inicia a comunicação serial
}
//endsetup

// ---------------------------------------------------------------------------------------------------------
//Início da Sessão LOOP

void loop(){

buttonState = digitalRead(botao1_pin);

// verifica se botão 1 foi pressionado

  if (buttonState == HIGH && flag1 == 0) {  // condição para o IF:
                                // botão apertado e flag em 0
    Serial.println ("botão apertado"); // envia essa mensagem pela serial
digitalWrite(ledPin1, HIGH); // acende o LED

//Como a tampa, por padrão de acionamento, é estar aberta, então o acionar do botão
    //é para a tampa ser fechada,
    //girando, assim, no sentido horário por EXATOS 90 graus.

    stepper.step(-50); //gira motor sentido horário
    flag1 = 1; // muda o valor da variável para que não entre novamente no IF
delay(1000); // aguarda 1s
  }

   /*
     if (digitalRead(botao1_pin) == LOW ){
stepper.step(-50); //gira motor sentido horário
        delay(1000); // aguarda 1s
     }
   */

//O próximo passo é esperar que o forno aquece e atinja 600 °C
//através de um sensor de temperatura, termopar tipo k

// verifica se botão 2 foi pressionado

    buttonState = digitalRead(botao2_pin);

    if (buttonState == LOW && flag2 == 1) { // condição para o IF:
                                // botão apertado e flag em 1
       Serial.println ("botão apertado");  // envia essa mensagem pela serial
digitalWrite(ledPin2, LOW); // desliga o LED

stepper.step(50); // gira motor sentido horário
       flag2 = 1; // muda o valor da variável para que não entre novamente no IF
```

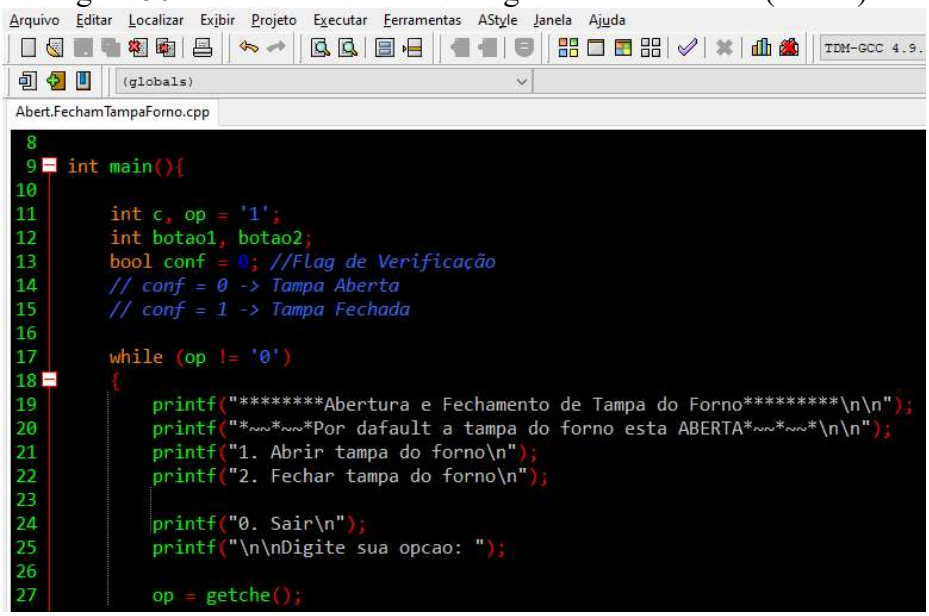
```
delay(1000); // aguarda 1s
    }

   /*
     if ( digitalRead(botao2_pin) == LOW) {
stepper.step(50); // gira motor sentido horário
       delay(10); // aguarda 10ms
     }
   */
}
```

Houve a necessidade de se usar um intertravamento entre os botões de acionamento, isto é, gerar uma segurança aos motores de passo de forma que eles não girassem mais vezes que o necessário e que girassem no sentido correto para que não ocasionasse ruptura das engrenagens e dos dispositivos. A implementação dos códigos fez uso de flags (usadas como sinal na programação para informar ao programa que uma determinada condição foi atendida) e com elas usou-se uma forma de verificação de estado para que, após esta, o procedimento seguinte pudesse ser realizado com segurança. Conseguiu-se, assim, controlar o motor de passo para a tampa do forno elétrico.

5.5.1.1 Código em Linguagem C

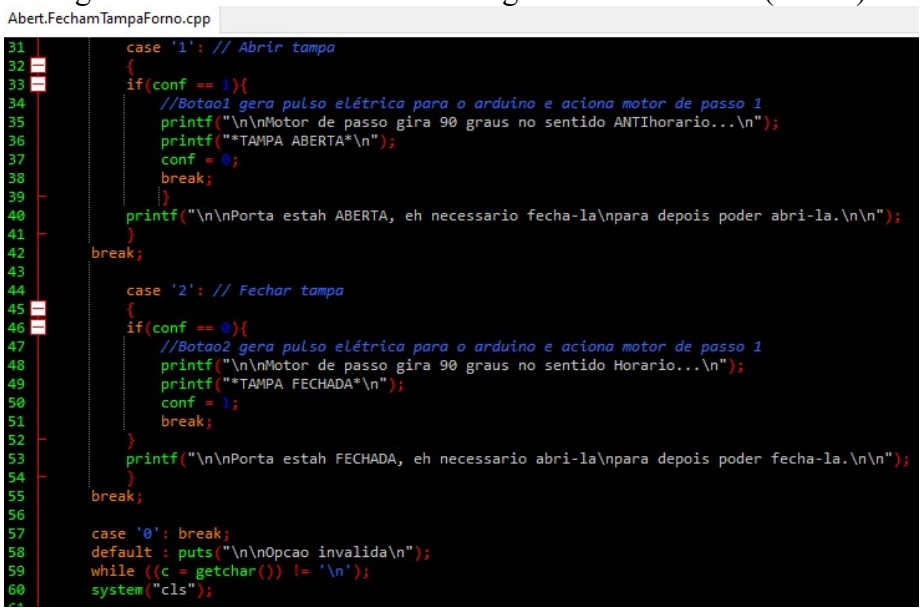
Código alternativo em linguagem C é mostrado abaixo e ilustrado nas figuras 50 e 51 na IDE DevC++. Esta linguagem possui uma grande quantidade de compiladores para quase todos os computadores e, dessa forma, o código implementado pode ser compilado e ter todas as instruções executadas com praticamente pouca ou nenhuma modificação. As grandes vantagens dessa linguagem é a portabilidade, o acesso fácil ao hardware e baixos requisitos de memória.

Figura 50 – IDE DevC++ com código alternativo em C (Parte1).



Fonte: Autor.

Figura 51 – IDE DevC++ com código alternativo em C (Parte2).



Fonte: Autor.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

/*************************************************/
/************** Programa Principal **************/
/*************************************************/

intmain(){

        int c, op = '1';
```

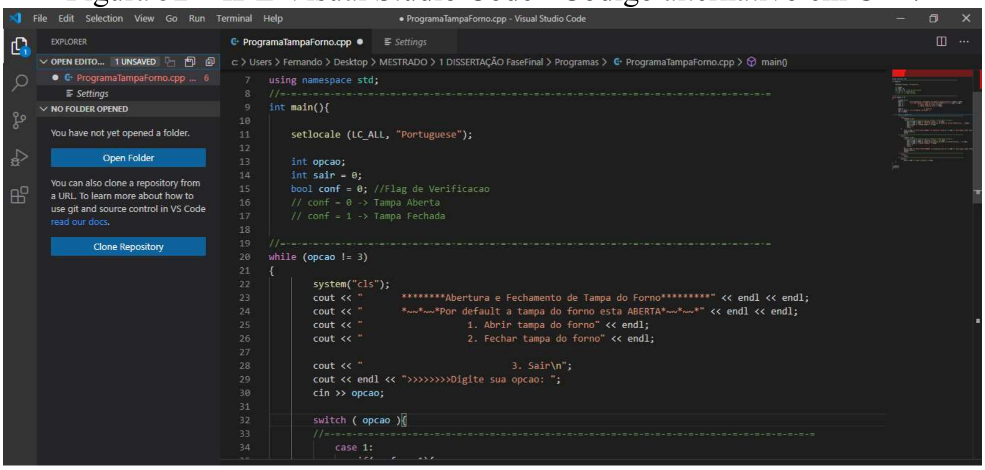
```
	int botao1, botao2;
	boolconf = 0; //Flag de Verificação
	// conf = 0 -> Tampa Aberta
	// conf = 1 -> Tampa Fechada

	while (op != '0')
	{
		printf("********Abertura e Fechamento de Tampa do Forno*********\n\n");
		printf("*~~*~~*Por dafault a tampa do forno esta ABERTA*~~*~~*\n\n");
		printf("1. Abrir tampa do forno\n");
		printf("2. Fechar tampa do forno\n");

		printf("0. Sair\n");
		printf("\n\nDigite sua opcao: ");

		op = getche();

	switch(op){

		case '1': // Abrirtampa
		{
		if(conf == 1){
			//Botao1 gera pulso elétrica para o Arduino e aciona motor de passo 1
			printf("\n\n\tAbrindo a tampa...\n");
			printf("\n\nMotor de passo gira 90 graus no sentido ANTIhorario...\n");
			printf("\n\t*TAMPA ABERTA*\n");
			conf = 0;
			break;
			}
		printf("\n\nPortaestah ABERTA, ehnecessario fecha-la\npara depois poder abri-la.\n\n");
		}
	break;

		case '2': // Fechartampa
		{
		if(conf == 0){
			//Botao2 gera pulso elétrica para o Arduino e aciona motor de passo 1
			printf("\n\n\tFechando a tampa...\n");
			printf("\n\nMotor de passo gira 90 graus no sentido Horario...\n");
			printf("\n\t*TAMPA FECHADA*\n");
			conf = 1;
			break;
		}
		printf("\n\nPortaestah FECHADA, eh necessario abri-la\npara depois poder fecha-la.\n\n");
		}
	break;
```

```
            case '0': break;
            default :puts("\n\nOpcao invalida\n");
            while ((c = getchar()) != '\n');
            system("cls");
            }
                    puts("\nPressione<ENTER> para continuar\n");
                    while ((c = getchar()) != '\n');
                    system("cls");
                    } /* Fim do while */
    return 0;
    }
```

5.5.1.2 Código em Linguagem C++

Código alternativo em linguagem C++ é mostrado abaixo e ilustrado na figura 52 na IDE Visual Studio Code. Cerca de 90% dos códigos para sistemas embarcados são escritos em C/C++ e apesar de ter uma codificação um pouco lenta e robusta, são gerados códigos de tempo de execução de forma compacta e rápida. Por possuir uma boa velocidade de tempo de execução de instruções, C++ é uma linguagem forte para implementação de softwares.

Figura 52 – IDE Visual Studio Code - Código alternativo em C++.



Fonte: Autor.

```
#include <iostream>
#include <cstdlib>
#include <locale>
#include <string>
#include <math.h>

using namespace std;

//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

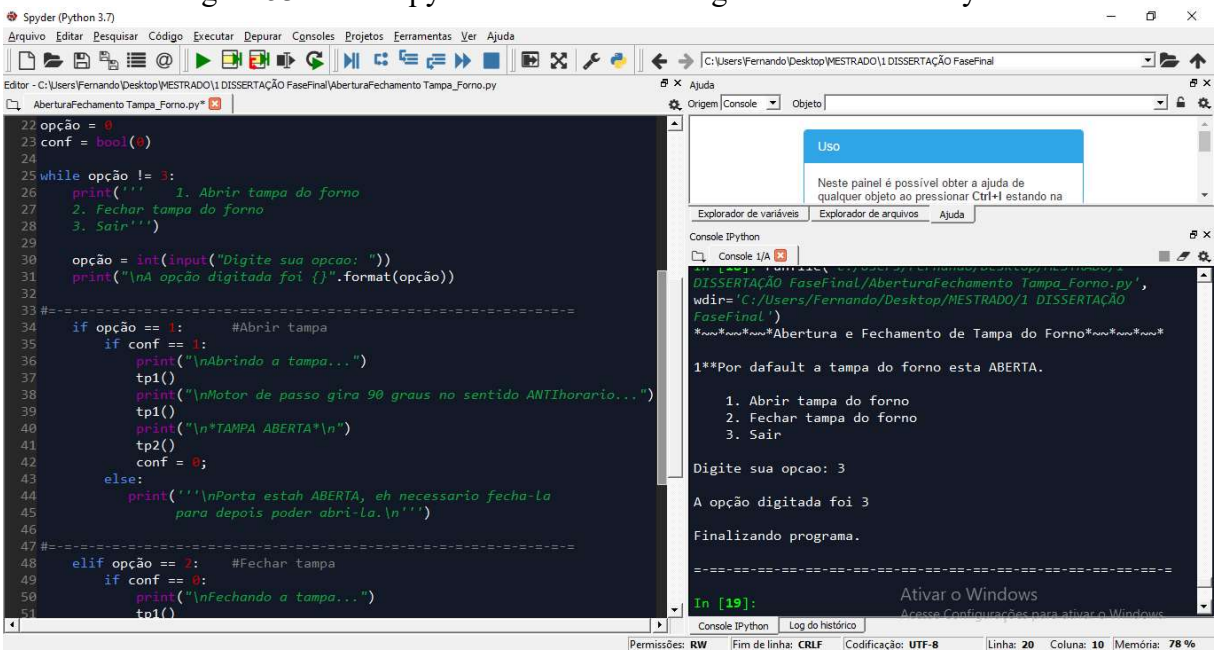
```
int main(){
        setlocale (LC_ALL, "Portuguese");
        int opcao;
        int sair = 0;
        bool conf = 0; //Flag de Verificacao
        // conf = 0 -> Tampa Aberta
        // conf = 1 -> Tampa Fechada

//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

while (opcao != 3)
{

system("cls");
cout << "*****Abertura e Fechamento de Tampa do Forno******" << endl << endl;
cout << "*~~*~~*Por default a tampa do forno esta ABERTA*~~*~~*" << endl << endl;
                cout << "                1. Abrir tampa do forno" << endl;
                cout << "                2. Fechar tampa do forno" << endl;

                cout << "                        3. Sair\n";
                cout << endl << ">>>>>>>>>Digite sua opcao: ";
                cin >> opcao;

//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

  switch ( opcao ){

        case 1:
          if(conf == 1){
          //Botao1 gera pulso eletrica para o arduino e aciona motor de passo 1
           cout << endl << endl << "Abrindo a tampa..." << endl;
    cout << endl << endl << "Motor de passo gira 90 graus no sentido ANTIhorario..." << endl;
            cout << endl << "*TAMPA ABERTA*\n" << endl;
            conf = 0;
            system("pause");
            break;
                    }
          cout << endl << "Porta estah ABERTA, eh necessario fecha-la para depois poder
abri-la.\n\n" << endl;
          system("pause");
          break;

//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

        case 2:
          if(conf == 0){
          //Botao2 gera pulso elï¿½trica para o arduino e aciona motor de passo 1
            cout << endl << endl << "Fechando a tampa..." << endl;
        cout << endl << endl << "Motor de passo gira 90 graus no sentido Horario..." << endl;
```

```
            cout << endl << "*TAMPA FECHADA*\n" << endl;
            conf = 1;
            system("pause");
            break;
        }
        cout << endl << "Porta estah FECHADA, eh necessario abri-la para depois poder
fecha-la.\n\n" << endl;
        system("pause");
        break;

//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

    case 3:
        cout << endl << "Programa sendo finalizado...\n";
        break;

//=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

    default:
            cout << endl << "Opcao invalida\n" << endl;
            system("pause");
  }
}
cin.get();
return 0;
}
```

5.5.1.3 Código em Linguagem Python

Código alternativo em linguagem Python é mostrado abaixo e ilustrado na Figura 53 na IDE Spyder – Anaconda. O uso dessa linguagem é impulsionado devido à redução da manutenção de um programa, suportando módulos e pacotes e reuso de códigos. Python é extremamente popular na comunidade científica e sua utilização cresce cada vez mais por sua ampla utilização em diversos sistemas operacionais, sistemas embarcados, além de auxiliar outras linguagens. Apesar da linguagem C/C++ dominar o desenvolvimento de sistemas embarcados, Python vem ganhando muita força neste campo, além de ser a linguagem que mais cresce quando o assunto é ciência de dados, inteligência artificial e computação incorporada. C/C++ apresenta alguns problemas de lentidão na escrita, sendo mais susceptível a erros, e Python vem com a grande vantagem por sua legibilidade, reduzindo erros

Figura 53 – IDE Spyder-Anaconda - Código alternativo em Python.



Fonte: Autor.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 30 19:18:20 2020
@author: Fernando
"""
import time

def tp1():
    time.sleep(3)

def tp2():
    time.sleep(5)

print("*~~*~~*~~*Abertura e Fechamento de Tampa do Forno*~~*~~*~~*")
print("\n1**Por default a tampa do forno está ABERTA.\n")

opção = 0
conf = bool(0)

while opção != 3:
    print("    1. Abrir tampa do forno
    2. Fechar tampa do forno
    3. Sair")

    opção = int(input("Digite sua opção: "))
    print("\nA opção digitada foi {}".format(opção))
```

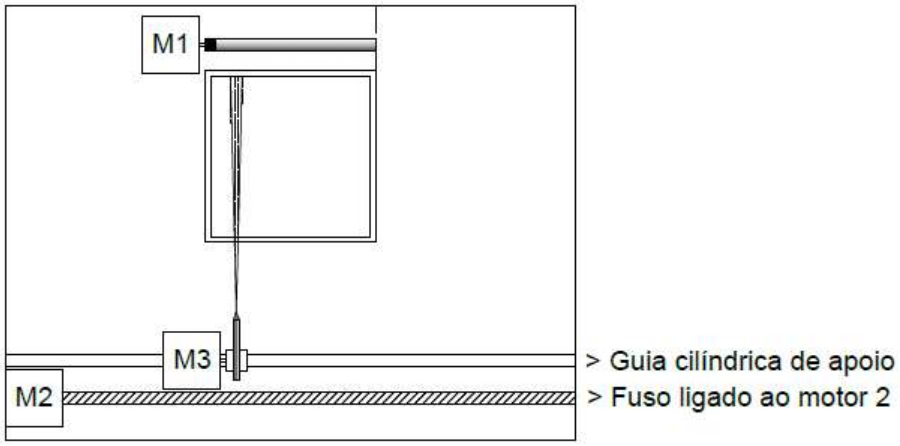
```
#=-=-=-=-=-=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

if opção == 1:     #Abrir tampa
        if conf == 1:
            print("\nAbrindo a tampa...")
            tp1()
            print("\nMotor de passo gira 90 graus no sentido ANTIhorário...")
            tp1()
            print("\n*TAMPA ABERTA*\n")
            tp2()
            conf = 0;
        else:
           print('"\nPorta está ABERTA, á necessário fecha-la
                 para depois poder abri-la.\n"')

#=-=-=-=-=-=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
    elif opção == 2:    #Fechar tampa
        if conf == 0:
            print("\nFechando a tampa...")
            tp1()
            print("\nMotor de passo gira 90 graus no sentido Horário...")
            tp1()
            print("\n*TAMPA FECHADA*\n")
            tp2()
            conf = 1
        else:
            print('"\nPorta está FECHADA, é necessário abri-la
                  para depois poder fechá-la.\n"')

#=-=-=-=-=-=-=-=-=-=-=-=-=-==-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
    elif opção == 3:    #Fechar programa
        tp1()
        print('\nFinalizando programa.\n')
    else:
        tp1()
        print("\n Opção Inválida. Digite novamente!")
    print('=-=' *20)
```

5.6 Circuito motores de passo e pistola aerógrafo

O circuito de principal impacto, circuito 2 na figura 48, na automação do forno elétrico e da câmara de aquecimento foi exatamente o controle dos motores de passo: um motor controlando o movimento do aerógrafo na horizontal e outro motor de passo controlando o movimento rotacional do aerógrafo.

A Figura 54 mostra como ficou o projeto da inserção da parte física na câmara de aquecimento.

Figura 54 – Projeto da parte física.

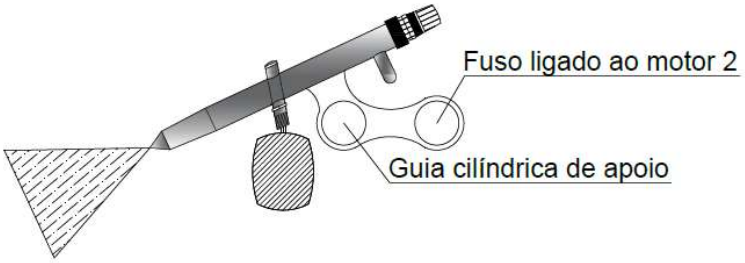


Fonte: Autor.

O código obedeceu a dois comandos de programação. A programação 1 foi percorrer o forno no eixo x com o M2, esquerda → direita, retornando à posição inicial, girando o aerógrafo 20° com M3 e novamente percorrendo o eixo x até finalizar a área 30x30cm do forno elétrico. A programação 2, o aerógrafo gira 60°, fazendo com que o spray percorra o "eixo y" de cima pra baixo e retornando, através do M3, com o M2, percorrendo o eixo x de 10 em 10 cm, continuando o ciclo até finalizar a área do forno 30x30cm.

O motor 3, responsável pelo aerógrado, ficaria acoplado a uma peça (Figura 55), será produzida em impressora 3D, que une a guia cilíndrica linear e o fuso, que juntamente com o motor 2, é o responsável pelo movimento na horizontal.

Figura 55 – Peça a ser produzida em impressora 3D.



Fonte: Autor.

O código a ser implementado que comande os 2 motores de passo está descrito abaixo. Este código serve de base para demais implementações futuras.

5.6.1 Códigos em Linguagem Wiring

5.6.1.1 Programa 1

```
//Programa para controlar 2 motores de passo pelo Arduíno, cada motor tem 2 botões,
//que fazem girar sentido horário e anti-horário numa velocidade prefixada.

#include <Stepper.h>

const int stepsPerRevolution = 200;
Stepper motor(stepsPerRevolution, 10,11,12,13);
Stepper motor1(stepsPerRevolution, 6,7,8,9);
int statusBotao1 = 0;
int statusBotao2 = 0;
int statusBotao3 = 0;
int statusBotao4 = 0;

const int botaoHorario = 1; //gira motor sentido horario
const int botaoAntihorario = 2; //gira motor sentido antihorario
const int botaoHorarioM1 = 4; // gira motor 1 sentido horario
const int botaoAntihorarioM1 = 5; //gira motor 1 sentido antihorario
int steps = 0;

void setup(){
  pinMode (botaoHorario, INPUT);
  pinMode (botaoAntihorario, INPUT);
  pinMode (botaoHorarioM1, INPUT);
  pinMode (botaoAntihorarioM1, INPUT);
  }
void loop(){
  statusBotao1 = digitalRead(botaoHorario);
  statusBotao2 = digitalRead(botaoAntihorario);
  statusBotao3 = digitalRead(botaoHorarioM1);
  statusBotao4 = digitalRead(botaoAntihorarioM1);
  if(statusBotao1 == HIGH){
    motor.step(1);
    motor.setSpeed(13);
  }
  if(statusBotao2 == HIGH){
    motor.step(-1);
    motor.setSpeed(13);
  }
  if(statusBotao3 == HIGH){
    motor1.step(1);
    motor1.setSpeed(95);
  }
  if(statusBotao4 == HIGH){
    motor1.step(-1);
```

```
    motor1.setSpeed(95);
  }
}
```

5.6.1.2 Programa 2

```
#include <AccelStepper.h>

int velocidade_motor = 100;
int aceleracao_motor = 100;
int sentido_horario = 0;
int sentido_antihorario = 0;
int numero = 0;

// Definicao pino ENABLE

int pino_enable = 10;

// Definicao pinos STEP e DIR

AccelStepper motor1(1,7,4);

void setup()
{
Serial.begin(9600);
pinMode(pino_enable, OUTPUT);

// Configuracoes iniciais motor de passo
  motor1.setMaxSpeed(velocidade_motor);
  motor1.setSpeed(velocidade_motor);
  motor1.setAcceleration(aceleracao_motor);

Serial.println("Digite 1, 2 ou 3 e clique em ENVIAR...");

} //Setup END


void loop()
{

  // Aguarda os caracteres no serial monitor
if (Serial.available() > 0)
  {
numero = Serial.read();
{
if (numero == '1')
     {
Serial.println("Numero 1 recebido - Girando motor sentido horario.");
digitalWrite(pino_enable, LOW);
```

```
sentido_horario = 1;
sentido_antihorario = 0;
      }

if (numero == '2')
      {
Serial.println("Numero 2 recebido - Girando motor sentido anti-horario.");
digitalWrite(pino_enable, LOW);
sentido_horario = 0;
sentido_antihorario = 1;
      }

if (numero == '3')
      {
Serial.println("Numero 3 recebido - Parando motor...");
sentido_horario = 0;
sentido_antihorario = 0;
       motor1.moveTo(0);
digitalWrite(pino_enable, HIGH);
      }
    }
  }

  // Move o motor no sentido horario
if (sentido_horario == 1)
  {
   motor1.moveTo(10000);
  }

  // Move o motor no sentido anti-horario
if (sentido_antihorario == 1)
  {
   motor1.moveTo(-10000);
  }
  // Comando para acionar o motor no sentido especificado
motor1.run();
}
```

5.6.1.3 Programa 3

Algumas configurações são modificadas para adaptar um código de uma impressora 3D para implementação de um código para a automação do forno de aquecimento.

```
#ifndef CONFIGURATION_H
#define CONFIGURATION_H
#define CONFIGURATION_H_VERSION 010109
```

```
//============================ Getting Started =============================
//============================ HANGPRINTER =============================

#define STRING_CONFIG_H_AUTHOR "Fernando Oliveira - Houdini"
#define SHOW_BOOTSCREEN
#define STRING_SPLASH_LINE1 SHORT_BUILD_VERSION
#define STRING_SPLASH_LINE2 WEBSITE_UR

// @section machine

#define SERIAL_PORT 0

//Determinação da Velocidade de Comunicação

#define BAUDRATE 115200

#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_RAMPS_14_EFB
//1 bico aquecido (Hotend), 1  ventilador (Fun) e 1 cama (bed)
#endif

// SEÇÃO MÁQUINA

#define POWER_SUPPLY 1
//1 = ATX oufontechaveada

#if POWER_SUPPLY > 0
// Enable this option to leave the PSU off at startup.
// Power to steppers and heaters will need to be turned on with M80.
//#define PS_DEFAULT_OFF

//#define AUTO_POWER_CONTROL       // Enable automatic control of the PS_ON pin
//  #if ENABLED(AUTO_POWER_CONTROL)
#define AUTO_POWER_FANS          // Turn on PSU if fans need power
#define AUTO_POWER_E_FANS
#define AUTO_POWER_CONTROLLERFAN
#define POWER_TIMEOUT 30
#endif
//============================ Endstop Settings ============================

// @sectionhoming

//Os fim de curso estarão localizados no mínimo do X e no mínimo do Y.
//Na verdade, um motor (M1) será usado para fazer a varredura no eixo x,com y = 0
//indo, p. ex, de x = 0 a x = 30(cm), e de x = 30 a x = 0
//Após isso, um segundo motor acoplado ao aerógrafo, gira 20° sentido anti horário,
//deslocando a projeção de saída do líquido (bico do aerógrafo) para um y = -3 (cm)
//percorrendo uma área de 30cm x 30cm
#define USE_XMIN_PLUG
#define USE_YMIN_PLUG
```

```
#define USE_ZMIN_PLUG
//Não há movimento no Eixo Z

// Enable pullup for all endstops to prevent a floating state
//Previnir flutuação no Arduino para ele não entender sinais errados
//de acionando de fim de curso
#define ENDSTOPPULLUPS
//#if DISABLED(ENDSTOPPULLUPS)

//#endif

// Mechanical endstop with COM to ground
//and NC to Signal uses "false" here (most common setup).
//Não é usado o NC, e sim o NO, por isso trocou-se o false por TRUE nas linhas abaixo
#define X_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
#define Y_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
#define Z_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
#define X_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
#define Y_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
#define Z_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of the endstop.
```

5.7 Contextualização com outros trabalhos

Muitas pesquisas abordam a automação de processos laboratoriais e industriais, podendo utilizar diferentes microcontroladores e linguagens de programação, mas o ponto primordial é a forma como a automação foi essencial na melhora da qualidade de um serviço ou produto. Estima-se que com os códigos produzidos e em funcionamento na câmara térmica, juntamente aos motores de passo, ocorrerão ganhos na produção dos filmes finos no forno.

A codificação torna-se importante não somente para implementar ações em fornos ou câmaras térmicas, mas também em um grande número de processos, como na elaboração de um sistema de automação para painéis fotovoltaicos controlados por arduino (Severo, 2017), mostrando a versatilidade deste microcontrolador em todos os ramos das energias renováveis. Neste estudo, houve um ganho de 28,13% em relação ao sistema fixo, o sistema rastreador, em que a placa fotovoltaica se reposiciona em relação ao sol, aumentando seu tempo de exposição, obteve melhores médias (tensão, corrente na carga, tensão de curto circuito, corrente de curto circuito, temperatura e irradiação solar sobre o painel) em relação ao convencional, comprovando que o sistema automatizado torna o sistema ainda mais eficiente.

Rodríguez et al (2015) mostraram que existem câmaras térmicas sem autofoco e propõem um sistema mecânico que permite o movimento automático da lente e sistemas de computador e eletrônicos. O resultado positivo encontrado por esses autores permite concluir que a automação de fornos elétricos, assim como a de câmaras térmicas podem ser programas a partir da utilização de arduino MEGA, arduino UNO, motores de passo e programação em C, C++ e Python. Além disso, para o controle do motor CC, pode ser utilizado o microcontrolador ATMEGA 16 e a técnica PWM (*Pulse Width Modulation*), técnicas que foram satisfatórias para um bom resultado com câmera térmica (Rodríguez et al., 2015).

Outro importante ponto na automatização dos processos é o abordado no trabalho de Ferreira (2019), que mostra como sendo um dos objetivos a redução do custo operacional, fazendo que com haja a possibilidade do reaproveitamento e remanejamento de servidores, alunos, operadores para atividades antes não exploradas a partir do IoT (*Internet Of Things* – Internet das coisas). IoT é uma tecnologia capaz de fazer o monitoramento de todo tipo de situação através da captação de dados, fazendo uso de microchips e sensores posicionados nos lugares corretos e evoluindo para uma interação máquina-a-máquina (M2M).

Sendo assim, com a automatização do processo, o objetivo se torna mais eficiente, e com os códigos apresentados nesse trabalho, já testados e prontos para a sua implementação física na câmara térmica, dar-se-ia início à melhoria da produção dos filmes finos, sendo o marco inicial para implementações e *updates* posteriores dos códigos.

Machado (2017) mostra um projeto de simulador modificado para que um determinado equipamento opere de forma automatizada por meio rotinas pré-programadas, escritas de acordo com os ensaios planejados para o simulador. Esses trabalhos presentes na literatura ratificam o pensamento de quão importantes e decisivos são os processos automatizados para a ciência e o meio acadêmico, uma vez que há o máximo de minimização de erros e uma maior padronização de produtos e serviços.

No estudo de Machado (2017) foi utilizado um controlador lógico programável (CLP), amplamente encontrado no meio industrial. Utilizou-se dois *field loggers* (registradores) e um supervisório, que monitora e comanda o simulador. No estudo desta dissertação utilizou-se o Arduino por sua compatibilização com o forno elétrico e devido aos movimentos que serão transmitidos para os motores de passo. Componentes como CLPs e registradores podem ser inseridos em futuros trabalhos, uma vez que a complexidade das ações realizadas pela pistola aerográfica pode aumentar.

O importante é ser traçado as rotinas do processo que serão realizadas, assim como no trabalho presente foi realizada o estudo dos ciclos que serão processados para a realização da

deposição dos filmes finos. Finalizado o processo de alimentação dos atuadores com os códigos implementados desse estudo, estima-se que, da mesma foram como o trabalho de Machado (2017), o sistema de automação e aquisição de dados torne-se um dos diferenciais do forno elétrico em sua produção de filmes finos no LAFFER/UFC.

Essa atualização permitirá realizar experimentos de forma controlada, assim como no processo de Machado (2017), que elevou o potencial de avaliar a combustibilidade de diferentes combustíveis nas condições transientes da desvolatilização e combustão, com controle de atmosfera, pressão e temperatura.

Grandes avanços tecnológicos são obtidos devido a automação dos processos. Martins da Silva (2019), em seu trabalho de Automatização do processo de secagem da parte ativa de transformadores de potência pelo método HOS – *hot oil spray*, apresenta um sistema desenvolvido com um microcontrolador de baixo custo, Arduino, e com materiais facilmente obtidos atualmente no mercado e criar um simples sistema de monitorização e controlo capaz de criar uma interface gráfica do sistema a comunicar com um microcontrolador, mostrando que a linha acadêmica de pesquisa de comunicação microcontrolador e interface gráfica aponta em alta e traz grandes ganhos à ciência e comunidade acadêmica.

6 CONCLUSÃO

O sistema de automação transformou o processo de fabricação laboratorial, trazendo mais rapidez e precisão. O estudo apresentou as principais etapas de um projeto de automação de uma câmara de aquecimento de um forno resistivo com finalidade de preparação de filmes finos através da técnica spray pirólise. Foi possível uma análise e entendimento prático das melhores configurações de todos os principais elementos construtivos que compõem um forno a resistência elétrica.

A criação dos códigos particionados facilitou a implementação individual de cada etapa, sem que uma necessariamente dependesse da outra. A primeira etapa do processo, com o código gerado para a abertura e fechamento da tampa do forno elétrico, foi elaborada em *Wiring* e dentro da IDE do Arduino. O segundo código apresentou mais desafios por unir mais dispositivos como termopares, display LCD, potenciômetro, cartão de memória, módulos para leitura de termopar, protoboard e o Arduino. Ainda assim foi concluída com sucesso sua implementação, captação das temperaturas e salvamento das informações.

Inicialmente houve uma inspiração em códigos de impressora 3D, porém, o número de variáveis presentes nesses códigos inviabilizada o código do aerógrafo, tornando-o muito complexo, apesar de que ainda assim foi uma alternativa para o controle dos motores de passo. Partiu-se então para a criação de códigos que fizessem a interação dos dois motores de passo. O código apresentou uma boa comunicação entre os dois motores de passo, um motor realizando primeiramente os movimentos no sentido horário e, após isso, realizando movimentos de rotação ou ainda a comunicação de um motor realizando primeiramente movimentos de rotação e depois movimentos na horizontal.

As linguagens de programação C, C++ e Python mostraram-se uma forte ferramenta como um intermediário de comunicação entre o usuário e o sistema embarcado trabalho. A velocidade de conexão e execução das instruções mostrou-se elevada, o envio de informações para o sistema embarcado permitiu que o usuário automatize todo o sistema de deposição de filmes finos.

O estudo mostrou-se viável para implementação na câmara térmica e forno. O sistema de automação e aquisição de dados é passível de atualizações, que poderão permitir a realização de ensaios sob uma gama diversificada de condições experimentais.

É importante salientar a contribuição que este trabalho trará com códigos em C, C++ e Python para a comunidade acadêmica uma vez que podem ser aproveitados em diversas searas da tecnologia. E ainda servirem de norte para implementações em outras áreas. Este trabalho

traz consigo a realização da automação em um meio cada vez mais ascendente e necessário, a inserção das energias renováveis na matriz energética mundial. A automatização dos processos potencializa toda conversão de energia e está se tornando uma ferramenta essencial e indispensável para a eficiência energética e geração de energia, seja na economicidade, na minimização de falhas humanas, no upgrade e aperfeiçoamento das formas de captar energia e transformá-las em energia elétrica

7 SUGESTÕES PARA TRABALHOS FUTUROS

Uma perspectiva para trabalhos futuros é a possibilidade de melhoria desses códigos e ainda a interação entre eles. Ainda pode-se desenvolver uma interface gráfica a qual, hospedada em um computador pessoal (PC), permite melhor interação do usuário com o sistema. Outros desafios propostos seriam: o controle ou medição da distância Z (distância entre o bico do atomizador e o substrato; medição da densidade da solução, da pressão do gás de arrasto, e ainda usar a medição da densidade para controlar a pressão do gás; (reprodutibilidade); controlar/medir o tempo de deposição; e ainda usar medição em tempo real, durante a deposição, da espessura do filme. Os experimentos foram realizados com sucesso, como a alimentação dos códigos no microcontrolador e o arduino enviou corretamente os dados para os motores.

REFERÊNCIAS

ACELERA 3D. **Drive motor de passo A4988 – RAMPS Arduino**. https://acelera3d.com, 2020. Disponível em: < https://acelera3d.com/produto/driver-a4988/>. Acesso em: 10 nov. 2019.

AGNIHOTRI, N. **Stepper Motors or Step Motors**. https://www.engineersgarage.com/, 2011. Disponível em: <www.engineersgarage.com/articles/stepper-motors?page=1>. Acesso em: 18 mai. 2018.

ALFACONNECTION, **Eletromagnetismo**. Alfaconnection. 2020. Disponível em: www.alfaconnection.pro.br/fisica/eletromagnetismo. Acesso em: 16 jan. 2020.

ANEEL – Agência Nacional de Energia Elétrica. **Banco de Informações de Geração da ANEEL**. Brasília, 2020. Disponível em: <http://www2.aneel.gov.br/aplicacoes/atlas/pdf/03-Energia_Solar(3).pdf > Acesso em: 7 jan. 2020.

ARDUIDO e CIA. **Como usar um drive A4988 com motor de passo Nema 17**. 2020. Disponível em: <https://www.arduinoecia.com.br/driver-a4988-com-motor-de-passo-nema-17>. Acesso em: 20 set. 2019.

ARDUINO & TECNOLOGIA. **Controle do motor de passo bipolar com o drive A4988**. 2020. Disponível em: < http://www.arduinoetecnologia.com.br/projetos/controle-de-motor-de-passo-bipolar-com-o-driver-a4988>. Acesso em: 26 out. 2019.

ARDUINO. **Software Arduino IDE** (versão 1.8.12). 2020. Disponível em: <https://www.arduino.cc/en/Main/Software>. Acesso em: 06 jan. 2020.

ATHOS ELECTRONICS. **Termopar – O que é e como funciona**. 2020. Disponível em: < https://athoselectronics.com/o-que-e-como-funciona-termopar/>. Acesso em: 10 jan. 2020.

AUTO CORE ROBÓTICA. **Placa Ramps 1.4**. 2020. Disponível em: < https://www.autocorerobotica.com.br/placa-ramps-14>. Acesso em: 22 set. 2019.

AVELAR, E., **Leitura e Datalogger de temperatura utilizando Termopar tipo K e MAX6675**. 2019. Disponível em: <https://easytromlabs.com/arduino/arduino-lab-19-leitura-e-datalogger-de-temperatura-utilizando-um-termopar-tipo-k-e-o-max6675/>. Acesso em: 04 jan. 2020.

BAKKER, B. **How to control a stepper motor with DRV8825 driver and Arduino**. Makerguides. 2019. Disponível em: < https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/#comments>. Acesso em: 15 dez. 2019

BANZI, M. **Primeiros passos com o arduino**. 1. ed. São Paulo: Novatec Ltda, 2010.

BARBEDO, J. G. A. **Automação Laboratorial**. 1. ed. Campinas: Embrapa Informática Agropecuária, 2012.

BÉLANGER, D.; DODELET, J. P.; LOMBOS, B. A.; DICKSON, J. I. Thickness dependence of transport properties of doped polycrystalline tin oxide films. **Journal of The Electrochemical Society**. v. 132, n. 6, p. 1398, 1985.

BORGES, L. E. **Python para Desenvolvedores**. Rio de Janeiro: Edição do Autor, 2010.

CABRAL, B. **Geração de Energia Solar Domiciliar e Empresarial cresce 44% no CE**. Diário do Nordeste. Fortaleza. 26 nov. 2018. Caderno Negócios. Disponível em: <https://diariodonordeste.verdesmares.com.br/editorias/negocios/geracao-de-energia-solar-domiciliar-e-empresarial-cresce-44-no-ce-1.2030800> Acesso em: 22 mai. 2019.

CESTILE, M. **Materiais elétricos: Compêndio de trabalhos**. Fornos Elétricos. Universidade Estadual do Oeste do Paraná. v. 4, n. 5, p. 331-351, 2012.

CID, A. S.; CORREA, T. Venturino: análise da variação de pressão em um tubo de Venturi utilizando Arduino e sensor de pressão. **Revista Brasileira de Ensino de Física**. v. 41, n. 3, 2019.

CONDIT, R.; JONES, D. W. Stepping motors fundamentals. **Microchip Technology Inc.** Publicação n. AN907, p. 1–22, 2004.

CONSTANDINOU, T. G. **Tudo sobre motores de passo**. http://w3.ufsm.br/, 2003. Disponível em: <http://w3.ufsm.br/fuentes/index_arquivos/step.pdf>. Acesso em: 15 out. 2019.

COUTO JR, M. A.; VIRTUOSO, G. H. F.; MARTINS, P. J. **Propriedades desejáveis a uma linguagem de programação: Uma Análise Comparativa entre as linguagens C, C++ e Java**. http://periodicos.unesc.net/index.php/sulcomp/article/viewArticle/796. ISSN 2359-2656. Congresso Sul Brasileiro de Computação. 2005.

CURVELLO, A.; BUENO, C.; SOUZA, F.; ROSSI, H.; PEREIRA, R.; LIMA, T. **Editorial: Linguagens de Programação para Sistemas Embarcados**. Embarcados. 2017. Disponível em: < https://www.embarcados.com.br/editorial-linguagens-para-sistemas-embarcados/>. Acesso em: 9 jan. 2020.

DEITEL, H. M.; DEITEL, P. J. **C++ Como Programar**. 3 ed. São Paulo: Bookman, 1098 p. 2001.

DI SOUZA, R. **Célula Fotovoltaica – O Guia Técnico Absolutamente Completo**. Fevereiro, 2017. Disponível em <https://blog.bluesol.com.br/celula-fotovoltaica-guia-completo/> Acesso em: 20 abr. 2020.

ELANGOVAN, E.; SHIVASHANKAR, S.A.; RAMAMURTHI, K. Studies on Structural and electrical properties of sprayed SnO_2: Sb films. **Journal of Crystal Growth**. v. 276, n. 1-2, p. 215-221, 2005.

ELECTRONOOBS. **Temperature PID controller – Arduino**. 2019. Disponível em: <http://electronoobs.com/eng_arduino_tut24.php>. Acesso em: 01 fev. 2020.

FALCONE, A. G., **Eletromecânica - Máquinas Elétricas Rotativas**, 5. ed. v. 3. São Paulo: Edgard Blücher Ltda, 1979.

FB Irrigação. **Kit Tubo de Venturi Valvulado**. fbirrigacao, 2020 Disponível em: <https://fbirrigacao.com.br/produtos/kit-tubo-de-venturi-valvulado>. Acesso em: 13 mar. 2020.

FERREIRA, J.; NOGUEIRA, G. R. G.; SANTOS, F. **Sistema de Sensoriamento Remoto da Câmara Fria do IFPI**. In: Escola Regional De Computação Aplicada À Saúde (Ercas), 2019, Teresina. Anais da VII Escola Regional de Computação Aplicada à Saúde. Porto Alegre: Sociedade Brasileira de Computação, p. 330-335, 2019

FRANCHI, C. M.; CAMARGO, V. L. A. **Controladores Lógicos Programáveis Sistemas Discretos**. 1. ed. São Paulo: Érica, 2008.

GARCIA, C. G.; MURAKAMI LHA, N. Y. Photoelectrochemical solar cells using $[(dcbH_2)_2RuLL']$,L,L' = substituted pyridines, as nanocrystalline TiO_2 sensitizers, **International Journal of Photoenergy**. v. 3, p. 131- 135, 2002.

GARCIA, C.G.; MURAKAMI, N.Y.; ARGAZZI, R.; BIGNOZZI, C. A. J. **Journal of Photochemistry and Photobiology A: Chemistry.** v. 115, p. 239-242, 1998.

GERALDO, V.; SCALVI, L. V. A.; MORAIS, E. A.; SANTILLI, C. V.; PULCINELLI, S. H. **Materials Research**, v. 6, n. 4, p. 451-456, 2003.

GRÄTZEL, M. A.; O'REGAN, B. Low-cost, high-efficiency solar cell based on dye-sensitized colloidal tio_2 films. **Nature**. v. 353, p.737–739, 1991.

GRAY, J. L. **Handbook of photovoltaic science and engineering**. 2. ed. England: John Wiley & Sons Ltd, 2003.

GREEN, M. A.; EMERY, K.; HISHIKAWA, Y.; WARTA, W.; DUNLOP, E. D.; Solar cell efficiency tables. **Progress in Photovoltaics: Research and Applications**. v. 20, p. 12-20. 2012.

GRENN, M. A. **Solar Cells, Operating Principles, Tecnhology, and System Applicatoins**. 1. ed. United States: Prentice-Hall, 1982.

GROOVER, M. **Automação industrial e sistemas de manufatura**. Tradução Jorge Ritter, Luciana do Amaral Teixeira, Marcos Vieira; revisão técnica José Hamilton Chaves Gorgulho Júnior. 3. ed. São Paulo: Pearson Prentice Hall, 2011.

GUERRA, L. N. A. **Uso de compensador PID no controle da taxa de variação de temperatura em um forno elétrico a resistência**, 2006, Monografia (Graduação em Engenharia Elétrica) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

HOFFMANN, W., PELLKOFER, T. Thin films in photovoltaics: Technologies and perspectives. **Thin Solid Films**. v. 520, n. 12, p. 4094 - 4100. 2011

HUBBARD, J. R. **Teoria e problemas de programação em C ++**. 2. ed. Porto Alegre: Bookman, 2003.

IEA. **International Energy Agency – IEA**. Key World Energy Statistics. 2019.

JONES, M. R. **Schematic illustration of a generic dye-sensitized solar cell**. 2009. Disponível em < https://upload.wikimedia.org/wikipedia/commons/f/fd/Dye_Sensitized_Solar_Cell_Scheme.png>. Acesso em: 10 fev. 2020.

JUSTI, M. A., **Automatização do controle de processo de refusão desolda "lead free" em uma linha de produção "SMD"**. Taubaté: Unitau, 2009. Dissertação – Universidade de Taubaté. Faculdade de Engenharia Mecânica. Curso de Automação Industrial e Robótica, Taubaté, 2009.

KOSOW, I. L. **Máquinas Elétricas e Transformadores**, 4. ed. Porto Alegre: Globo, 1982.

LEMOS, M. **Conheça os shields e incremente seu arduino com eles**. Fazedores. 2013. Disponível em < https://blog.fazedores.com/conheca-os-shields-e-incremente-seu-arduino-com-eles/>. Acesso em: 17 jan. 2020.

LIMA, F. M. **Deposição de Dióxido de Estanho-Flúor (Sno2:F) em Substrato Transparente para Uso em Células Fotoeletroquímicas**. 2013. Dissertação (Pós-Graduação em Engenharia Mecânica) – Universidade Federal do Ceará, Fortaleza, Ceará, 2013.

LUGLI, A. B.; SANTOS, M. M. D. **Redes Industriais para Automação Industrial: AS-I, Profibus e Profinet**. 2 ed. São Paulo: Érica, 2019.

MACHADO, A. da S. **Automação de simulador de combustão para avaliação dos fenômenos transientes durante a desvolatilização e combustão de carvões para injeção em altos-fornos**. - 2017. 133 f., enc.: il. Tese - Universidade Federal de Minas Gerais, Escola de Engenharia.

MADEIRA, D. **Termopar tipo K + MAX6675 – Medindo Temperatura**. 2018. Vida de Silício. Disponível em: < https://portal.vidadesilicio.com.br/termopar-tipo-k-max6675/>. Acesso em: 10 jan. 2020.

MAIA Jr., P. H. F. **Obtenção de um filme fino de FTO pela técnica de Spray-pirólise e método Sol-Gel, para utilização em células solares orgânicas**. 2015. Dissertação (Pós-Graduação em Engenharia Mecânica) – Universidade Federal do Ceará, Fortaleza, Ceará, 2015.

MARTINS DA SILVA, M. M. **Automatização do processo de secagem da parte ativa de transformadores de potência pelo método HOS – hot oil spray**. Departamento de Engenharia Eletrotécnica, Dissertação de Mestrado em Engenharia Eletrotécnica – Sistemas Elétricos de Energia. 2019.

MARUYAMA, K.; TOBATA, J. Indium-Tin Oxide Thin Films Prepared by Chemical Vapor Deposition from Metal Acetates. **Journal of Applied Physics**. v. 68, p. 4282-4285, 1990.

MCROBERTS, M. **Arduino básico**. 1. ed. São Paulo: Novatec Ltda. 2011.

MEIQIN, M.; JIANHUI, S.; CHANG, L.; KAI, P.; GUORONG, Z.; MING, D. Researchand Development of Fast Field Tester for Characteristics of Solar Array. **IEEE Canadian Conference on Electrical and Computer Engineering**. p. 1055 - 1060, 2009.

MENDONÇA, H. S. **SPI e I2C**. Página da atividade docente e de investigação. 2020. Disponível em: < https://paginas.fe.up.pt/~hsm/docencia/comp/spi-e-i2c/>. Acesso em: 20 jan. 2020.

MOHAN, N.; UNDELAND, T. M.; ROBBINS, W. P. **Power Electronics. Converters, Applications and Design**. 3. ed. Minneapolis: John Wiley & Sons. 2003.

MONK, S. **Programação com Arduino - Começando com sketches**. 2. ed. Porto Alegre: Bookman Ltda. 2013.

MORAES, C. C.; CASTRUCCI, P. de L. **Engenharia de automação industrial**. 2. ed. Rio de Janeiro: LTC. 2010.

MURTA, G. **Guia completo do display LCD – Arduino**. Eletrogate. 2018. Disponível em: <https://blog.eletrogate.com/guia-completo-do-display-lcd-arduino/>. Acesso em: 15 jan. 2020.

NORONHA, J. F. V. **Obtenção e caracterização de filmes SnO2 depositados em vidro borosilicato por silk-screen modificado: SnCl2.2H2O como precursor**. 2007. Dissertação (Mestrado em Engenharia Mecânica) – Escola de Engenharia Universidade Federal do Rio Grande do Norte, Natal, 2007.

OLIVEIRA, E. **Como usar com Arduino – Módulo MAX6675 – Termopar tipo K (0° a 800°C)**. 2018. Disponível em: < https://blogmasterwalkershop.com.br/Arduino/como-usar-com-Arduino-modulo-max6675-termopar-tipo-k-0o-a-800oc/>. Acesso em 20 jan. 2020.

PANOSSO, A. R. **Linguagem de Programação C++**. Universidade Estadual Paulista Júlio de Mesquita Filho. Faculdade de Ciências Agrárias e Veterinárias. p. 48-100, 2019.

PATSKO, L. F., **Tutorial de Controle de Motor de Passo**. Maxwell Bohr – Instrumentação Eletrônica, Pesquisa e Desenvolvimento de Produtos. Londrina-PR. 2006.

PEREIRA, S. do L. **Linguagem C++**. São Paulo: FATEC. 1999.

QUASCHNING, Volker. **Understanding Renewable Energy Systems**. 1. ed. London: Earthscan, 2005.

QUEIROZ, R. A. de A. **Motores de Passo**. 2002. Disponível em: <https://docplayer.com.br/4118210-Motores-de-passo-ricardo-alexandro-de-andrade-queiroz.html>. Acesso em: 02 nov. 2019.

QUEIROZ, W. R. de O.; SOUSA, W. Q. A Importância da Plataforma Arduino no Meio Acadêmico. **Revista Científica Multidisciplinar Núcleo do Conhecimento**. Ano 3, Ed. 8, v. 12, p. 123-133, 2018.

REGADAS, J. J. L. M. **Otimização de uma impressora 3D Delta e desenvolvimento da impressão simultânea de 3 cores**. 2017. Dissertação. (Mestrado Integrado em Engenharia Mecânica). Porto, Portugal. Faculdade de Engenharia Universidade do Porto, 2017.

RIBEIRO, A. F. **Projeto e montagem da Câmara de aquecimento para produção de filmes finos de Sno2**. 2019. Monografia (Graduação em Engenharia Mecânica) – Universidade Federal do Ceará, Ceará, 2019.

RODRIGUES, C. H. M. **Implementação de sistema de spray pirólise com movimento equatorial para deposição de filmes cerâmicos derivados de zircônia estabilizada com ítria**. 2008. 131 f. Tese (Engenharia e Ciência dos Materiais), Campos dos Goytacazes – RJ. Universidade Estadual do Norte Fluminense Darcy Ribeiro CCT-LAMAV. 2008

RODRIGUES, M. **Tutorial: Comunicação SPI (Serial Peripheral Interface) com Arduino**. Laboratório de Garagem. 2016. Disponível em: < http://labdegaragem.com/profiles/blogs/tutorial-comunica-o-spi-serial-peripheral-interface-com-arduino>. Acesso em: 18 jan. 2020.

RODRÍGUEZ E.; TOAPANTA A.; RODAS A. Sistema de enfoque automático para una cámara térmica, usando procesamiento de imágenes en MATLAB. **Revista Politécnica – Febrero**. v. 35, n. 2, 2015.

RÜTHER, R. **Edifícios Solares Fotovoltaicos: o potencial da geração solar fotovoltaica integrada a edificações urbanas e interligada à rede elétrica pública no Brasil**. Florianópolis: LABSOLAR. 2004.

SAGA, T.; Advances in crystalline silicon solar cell technology for industrial mass production. **NPG Asia Mater**. v. 2, p. 96-102, 2010.

SAWIN, J. L.; SVERRISSON, F.; Renewables 2019: Global Status Report. **REN21 Renewables Now**. Paris: REN21Secretariat. v. 1, p. 41, 2019.

SEVERO, K. **Prototipação e Elaboração de um Sistema de Automação para Painéis Fotovoltaicos Controlados por Arduino**. Anais do 9º SALÃO INTERNACIONAL DE ENSINO, PESQUISA E EXTENSÃO – SIEPE. Universidade do Pampa-RS. Campus Santana do Livramento. 2017.

SILVA, G. M. **Termopares – Dispositivos utilizados para medir temperatura**. 2018. Setúbal/IPS – Escola Superior de Tecnologia de Setúbal. Disponível em: <http://www.dem.feis.unesp.br/maprotec/termopares-dispositivos-utilizados-para-medir-temperatura.pdf>. Acesso em: 06 jan. 2020.

SOUSA, F. G. **SPI**. Mundo Projetado, Site para aprendizes e hobbistas de eletrônica. 2020. Disponível em: <http://mundoprojetado.com.br/spi/>. Acesso em: 09 fev. 2020.

SOUZA, A. R.; PAIXÃO, A. C.; UZÊDA, D. D.; DIAS, M. A.; DUARTE, S.; AMORIM, H. S. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. **Revista Brasileira de Ensino de Física**. Rio de Janeiro-RJ, v. 33, n. 1, p.1702. 2011.

SOUZA, F. **Placa Arduino Mega 2560**. Embarcados. 2017. Disponível em: <https://www.embarcados.com.br/arduino-mega-2560/>. Acesso em: 26 set. 2019.

SUNERGIA. **Energia Solar Fotovoltaica | Futuro mais Sustentável**. 2018. São Paulo. Disponível em: <https://sunergia.com.br/blog/energia-solar-fotovoltaica-futuro-mais-sustentavel/>. Acesso em: 12 dez. 2019.

SWIEGERS, W.; ENSLIN, J. H. R. An integrated maximum power point tracker for photovoltaic panels. **IEEE International Symposium on Industrial Electronics. Proceedings. ISIE'98**. v. 1, p. 40–44, 1998.

THANGARAJU, B. Structural and electrical studies on highly conduction spray deposited fluorine and antimony doped SnO_2 precursor. **Thin Solid Films**. v. 402, n. 2, p.71-78. 2002.

THOMSEN, A. **Controlando LCD 16x2 com Arduino. Felipeflop**. 2011. Disponível em: <https://www.filipeflop.com/blog/controlando-um-lcd-16x2-com-arduino/>. Acesso em: 24 jan. 2020.

TURK. **Fim de Curso, como funciona**. 2020. Disponível em: https://www.turk.com.br/paginas/Chave-fim-de-curso/fim-de-curso-como-funciona.php. Acesso em: 05 fev. 2020.

UNIVERSIDADE FEDERAL DO CEARÁ. Biblioteca Universitária. **Guia de normalização de trabalhos acadêmicos da Universidade Federal do Ceará**. Fortaleza, 2013.

USINAINFO, Eletrônica & Robótica. **Motor de Passo**. 2019. Disponível em: <www.usinainfo.com.br/motor-de-passo-472> Acesso em: 10 nov. 2019

WEIHUA, Y.; LAN, C.; MOGEN, X.; YUSHENG, H. Design of a Step-Motor Control System Based on FPGA. **IEEE Transactionson Automatic Control**, p. 4–874–4–877, 2007.

YELLA, A. LEE, H-W; TSAO, H. K.; YI, C.; CHANDIRAN, A. K.; NAZEERUDDIN, Md.K.; DIAU, E. W.-G.; YEH, C.-Y.; ZAKEERUDDIN, S. M.; GRÄTZEL, M. Porphyrin-sensitized solar cells with cobalt (ii/iii)-based redox electrolyte exceed 12 percent efficiency. **Science**. v. 334, p. 629–633, 2011.