



**SÃO PAULO**  
GOVERNO DO ESTADO

## **Ensino Técnico com Habilitação Profissional em Eletrônica**

**Ana Carolina Blasque Pereira**

**Ana Clara Soares Mendes**

**Felipe Mascarenhas Montresol Teles**

**Gustavo César Pereira**

**Isabela Santos Silva**

**Isabella Eloy Pereira**

**Desenvolvimento de suporte para medidas  
fotovoltaicas para células solares de Perovskita**

**Santo André - SP**

**2024**

**Ana Carolina Blasque Pereira**

**Ana Clara Soares Mendes**

**Felipe Mascarenhas Montresol Teles**

**Gustavo César Pereira**

**Isabela Santos Silva**

**Isabella Eloy Pereira**

## **Desenvolvimento de suporte para medidas fotovoltaicas para células solares de Perovskita**

Projeto de Trabalho de Conclusão de Curso apresentado ao Curso Técnico com Habilitação Profissional em Eletrônica da Etec Júlio de Mesquita, sob orientação da professora Adriana Mariko e Egmar Accetto, como parte dos requisitos para a obtenção do título de Técnico em Eletrônica

**Santo André - SP**

**2024**

## **AGRADECIMENTOS**

Gostaríamos de expressar nossa sincera gratidão à nossa coordenadora pelo apoio, orientação e dedicação ao longo de todo o desenvolvimento deste trabalho. Seu comprometimento com nossa formação foi essencial para a conclusão deste TCC.

Agradecemos também aos professores que, com seus valiosos conhecimentos e orientações, contribuíram de forma significativa para enriquecer nosso trabalho.

Nosso reconhecimento se estende aos membros do grupo, pelo apoio mútuo e pela colaboração em cada etapa deste processo. Manifestamos ainda nossa gratidão às instituições e pessoas que contribuíram direta ou indiretamente para a realização deste estudo.

A todos, nossos sinceros agradecimentos!

## LISTA DE ILUSTRAÇÕES

Figura 1 – Arduino Mega 2560 .....	12
Figura 2 – Motor de Passo 28BYJ-48 e Módulo de Motor de Passo Driver ULN2003.....	14
Figura 3 - Arduino Nano.....	14
Figura 4 - Encoder KY040 .....	15
Figura 5 - Solid States .....	17
Figura 6 - Célula de Perovskita .....	18
Figura 7 - Placa De PCI.....	18
Figura 8 - Montagem no Protoboard.....	19
Figura 9 - Início da Montagem.....	19
Figura 10 - Placa com Suporte Finalizado.....	20
Figura 11 – Esquema elétrico .....	21
Figura 12 - Programação.....	22
Figura 13- Caixinha Final.....	28
Figura 14- Tampa Da caixa.....	29
Figura 15 – Suportes Internos da célula.....	29

## LISTA DE TABELAS

Tabela 1 - Equipe Técnica .....	9
Tabela 2 - Materiais (Máscara de sombra) .....	11
Tabela 3 - Materiais (PCI) .....	11
Tabela 4 - Custos dos componentes.....	23
Tabela 5 - Custo Placa de PCI.....	24
Tabela 6 - Custos Gerais .....	24

## RESUMO

A energia fotovoltaica, uma das fontes renováveis mais promissoras, tem se destacado especialmente com o uso de células solares de perovskita, que oferecem alta eficiência de conversão e custo de produção inferior aos compostos convencionais. Estas células, compostas por um material híbrido de estrutura cristalina ABX<sub>3</sub>, alcançaram eficiências comparáveis às células de silício, tornando-se relevantes na última década.

Apesar de seu potencial, ainda há desafios a serem enfrentados, especialmente no desenvolvimento de suportes adequados para medir a eficiência e estabilidade das células de perovskita. A medição precisa requer um fornecedor de energia solar estável, software para análise de dados e um suporte robusto.

O projeto proposto visa desenvolver um suporte automatizado combinado com uma máscara de sombra para melhorar a medição das células solares de perovskita, contribuindo para o avanço científico na área e oferecendo uma solução acessível para aquisição.

**Palavras-chave:** Células solares de perovskita; Medidas fotovoltaicas; Desenvolvimento de suporte; Eficiência de conversão de energia; Energia solar; Tecnologia de energia renovável; Estabilidade de células solares; Sustentabilidade.

## SUMÁRIO

1	INTRODUÇÃO .....	8
2	Dados do projeto .....	9
2.1	Equipe Técnica.....	9
3	CRONOGRAMA DAS ATIVIDADES .....	10
4	DESENVOLVIMENTO DO PROJETO.....	10
4.1	Objetivo Geral .....	10
4.2	Objetivos Específicos .....	10
5	MATERIAIS .....	10
5.1	Descrição dos Materiais .....	12
5.1.1	Arduino Mega .....	12
5.1.2	Motor de Passo (28BYJ-48).....	13
5.1.3	Módulo do Motor de Passo (Driver ULN2003) .....	13
5.1.4	Arduino Nano .....	14
5.1.5	ENCOLDER KY-040.....	15
5.1.6	SOLID STATE RELAY CPC1017N .....	16
5.2	Processos de Montagem.....	19
5.3	Princípio de Funcionamento.....	20
6	Programação .....	22
7	CUSTOS.....	23
8	RESULTADOS OBTIDOS.....	24
9	CONCLUSÃO .....	26
10	REFERÊNCIAS .....	27
11	ANEXOS .....	28
11.1	Cronograma de Atividades .....	28
11.2	Desenhos .....	28
11.3	Programação.....	30

## 1 INTRODUÇÃO

A energia fotovoltaica, ou energia solar, é uma das fontes de energia renovável mais promissoras e sustentáveis no presente momento. Dessa forma, dentro de seu campo de estudo, as células solares de perovskita emergiram como uma tecnologia revolucionária, devido à sua alta eficiência de conversão de energia e ao custo relativamente baixo de produção em relação aos compostos condutores já utilizados na indústria.

Tais dispositivos fotovoltaicos de perovskita são nomeados a partir de sua estrutura cristalina, composta por um material híbrido orgânico-inorgânico de estrutura cristalina que tem fórmula genérica ABX. Por isso, essas células possuem um destaque significativo na última década, uma vez que alcançaram eficiências de conversão de energia que rivalizam com as de células solares de silício tradicionais a partir dos resultados obtidos nos últimos estudos em diferentes laboratórios

Contudo, apesar do seu potencial, ainda existem diversas instigações a serem pesquisadas e aprimoradas no desenvolvimento de células solares de perovskita, sendo, por exemplo, a emergente necessidade em desenvolver suportes adequados para obtenção de medidas de aproveitamento dos dispositivos fotovoltaicos, uma vez que, esses suportes são de suma importância para medir com precisão a eficiência e a estabilidade de tais dispositivos.

Dessa forma, analisando o procedimento de medida em diferentes laboratórios, se tem um padrão de etapas a serem realizadas, sendo elas, a utilização de um fornecedor de energia solar que permita uma estabilidade em relação a condições externas para uma maior padronização de resultados da medida, a aplicabilidade de um software para ler e interpretar os dados obtidos da medida em formato de gráfico e, ainda, um suporte de sustentação do dispositivo para o processo de medição.

Em consonância disto, o projeto escolhido pelo grupo tem como principal objetivo o desenvolvimento de um suporte automatizado em conjunto com uma máscara de sombra para a medição das células. A fim de promover um avanço na área científica relacionada a medição de células de perovskita, além da criação de um suporte mais acessível para compra.

## 2 DADOS DO PROJETO

O projeto consiste no desenvolvimento de um dispositivo rotativo, mais conhecido como máscara de sombra, que se encontrará sobre o suporte de células de perovskita, que permite o experimento de medição da sua capacidade;

Nosso protótipo executará as principais funções designadas para uma máscara de sombras, sendo seu principal objetivo cobrir as outras células enquanto uma é exposta a energia fotovoltaica.

Diante disso, a sua configuração e movimentação será toda automatizada, programada em C++, esta programação será controlada pelo microcontrolador Arduino, peça-chave em nosso protótipo, que estará presente no suporte da célula

Paralelamente a isso, a carcaça de proteção das células será projetada em impressão 3D, ela possuirá uma abertura na parte superior que permitirá o encaixe da máscara de sombra rotativa.

### 2.1 Equipe Técnica

COMPONENTES	FUNÇÃO	DESCRIÇÃO
Ana Carolina Blasque Pereira	Documentação, parte mecânica	Documentação Final Dispositivo Suporte
Ana Clara Soares Mendes	Programação, parte mecânica	Dispositivo Suporte e Execução da Programação
Felipe M. M. Teles	Programação, documentação e montagem	Documentação Final, Montagem do Protótipo, Execução da Programação
Gustavo César P.	Montagem, programação	Montagem do Protótipo, Execução da Programação
Isabela Santos S.	Design do suporte	Desenhos do projeto
Isabella Eloy	Documentação, Design do suporte	Desenhos do projeto, Documentação Final

*Tabela 1 - Equipe Técnica*

### **3 CRONOGRAMA DAS ATIVIDADES**

Anexo ao Final.

### **4 DESENVOLVIMENTO DO PROJETO**

O principal objetivo do projeto é automatizar a medição das células de Perovskita, facilitando a interpretação de gráficos e resultados obtidos do experimento, além disso é focado no desenvolvimento de um projeto acessível e que garanta um resultado eficiente.

#### **4.1 Objetivo Geral**

Aprimorar as medidas das células voltaicas de Perovskita, de forma que obter resultados mais precisos ao aplicar o sistema de medida.

#### **4.2 Objetivos Específicos**

- Maior Automatização do Sistema;
- Facilitar o Procedimento de Medida ao selecionar cada contato e, ao mesmo tempo, na identificação do contato selecionado;
- Baixa Resistencia (próximo a zero) aplicada sobre os contatos do dispositivo;
- Redução de ruídos no procedimento de medida.

### **5 MATERIAIS**

Ao definir o tema, foi necessária uma análise de quais materiais seriam melhores para o protótipo. Após essa decisão visamos em focar primeiramente na parte relacionada ao suporte das células, sendo assim os materiais foram analisados para que garantisse uma qualidade adequada e duradoura. A partir disso, os materiais foram comprados conforme a lista a seguir:

Nome do componente	Especificação	Classificação
Arduino Mega 2560	ATmega2560	Microcontrolador
Filamento Preto	PLA	Filamento
Protoboard	830 pontos	Placa de Ensaio
Kit de Jumpers	(Macho-macho; macho-fêmea)	Cabo conector
Botão	Botão	Chave de Acionamento
Módulo Motor de Passo	Driver ULN2003	Módulo
Motor de Passos	28BYJ-48	Motor

*Tabela 2 - Materiais (Máscara de sombra)*

Nome do componente	Especificação	Classificação	Dimensões
Arduíno Nano V3	ATmega328	Microcontrolador	45 x 18mm
Relé de Estado Sólido CPC1017N	Polo único, Canal NA (1-Form-A), tipo MOSFET	Relé para chaveamento	10mm x 10mm
Encoder KY040	20 pulsos	Encoder	32 x 19 x 30mm
LEDs	Amarelo	Fotodiodo	5mm
Resistores	470 $\Omega$	Resistores	3,2mm x 1,6mm
Resistores SMD	470 $\Omega$	Resistor, 1206	3,2 mm x 1,6 mm
Protoboard	830 furos	Placa de ensaio	10C x 10L x 10A milímetros
Jumpers	(Macho-macho; macho-fêmea)	Cabo conector	Largura do conector: 2,54mm
Multímetro	Marca: BWX	Current Tester	15 x 10,5 x 4 centímetros
Ferro de Solda	-	-	-
Estanho	Liga 63/37	Estanho para solda	1mm
Plug para fio	Macho	Banana	4mm
Conector	2 vias	Conector	11mm x 14mm

*Tabela 3 - Materiais (PCI)*

## 5.1 Descrição dos Materiais

### 5.1.1 Arduino Mega

O Arduino Mega 2560 é uma placa microcontroladora que atua na movimentação da máscara de sombra.

**Tamanho:** Não é tão compacto como outros modelos Arduino, utilizado em projetos de grandes dimensões.

**Microcontrolador:** Usa o ATmega2560, operando a 16 MHz.

**Entradas/Saídas:** 54 pinos digitais (15 PWM), 16 entradas analógicas e 4 portas de comunicação serial UART.

**Memória:** 256 KB de memória Flash (8 KB utilizados pelo bootloader), 8 KB de SRAM e 4 KB de EEPROM.

**Conectividade:** Comunicação via USB Tipo-B, compatível com I<sup>2</sup>C, SPI e UART e Interface ICSP para gravação direta.

**Portabilidade:** Alimentação via USB ou fonte externa (tensão recomendada: 7-12V). Relativamente maior e menos portátil, mas com mais recursos de I/O.

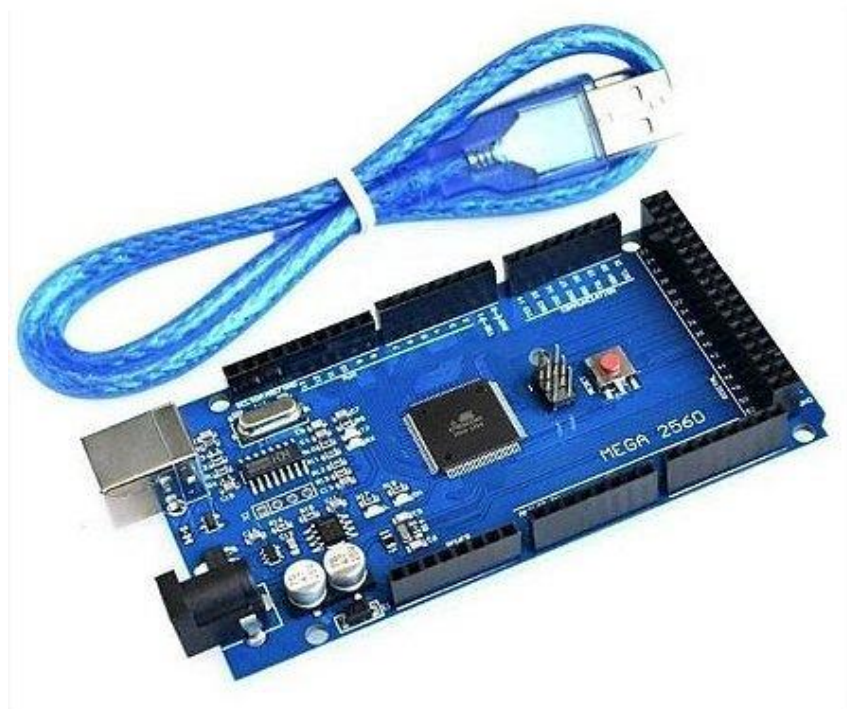


Figura 1 – Arduino Mega 2560

### 5.1.2 Motor de Passo (28BYJ-48)

O Motor de Passo (28BYJ-48) é um motor de passo compacto e econômico, amplamente utilizado em projetos de eletrônica e automação devido à sua precisão e simplicidade.

**Tensão de operação:** 5V DC.

**Resolução:** 64 passos por revolução no eixo do rotor. Com redução interna de engrenagem (64:1), alcança 4096 passos para uma rotação completa do eixo de saída.

**Consumo de corrente:** Aproximadamente 240 mA por bobina.

**Número de fases:** 4 fases (tipo unipolar), com bobinas independentes.

**Aplicações comuns:** Impressoras, dispositivos automáticos, robótica e sistemas de controle de posição.

**Funcionamento:** O motor utiliza quatro bobinas controladas em sequência para gerar um campo magnético rotativo que move o rotor. Essa sequência é controlada eletronicamente para determinar a direção e o número de passos.

### 5.1.3 Módulo do Motor de Passo (Driver ULN2003)

O ULN2003 é um módulo driver projetado para controlar motores de passo unipolares como o 28BYJ-48.

**Chip controlador:** ULN2003, que contém 7 canais de saída (4 utilizados para motores de passo).

**Compatibilidade:** Pode ser controlado por microcontroladores como Arduino, ESP32, PIC, etc.

**Proteção interna:** Diodos de proteção contra tensões reversas, protegendo contra picos de tensão gerados pelas bobinas do motor.

**Tensão de operação:** Suporta tensões de entrada de até 30V.

**Corrente máxima por canal:** 500 mA (por saída).

**Conexões:** Pinos IN1 a IN4 para conexão com o microcontrolador (entrada). Conector de 5 pinos para ligação ao motor de passo. LEDs indicativos mostram o acionamento de cada bobina (saída).

**Vantagens do módulo:** Facilita a interface entre o motor e o microcontrolador, reduz a necessidade de circuitos adicionais para controle de corrente, permite maior estabilidade e controle confiável.

**Aplicações comuns:** Controle de motores de passo em projetos de automação e robótica.



*Figura 2 – Motor de Passo 28BYJ-48 e Módulo de Motor de Passo Driver ULN2003*

#### **5.1.4 Arduino Nano**

O Arduino Uno é uma placa microcontroladora que serve como o "cérebro" do nosso projeto.

**Tamanho Compacto:** Ideal para projetos com espaço limitado.

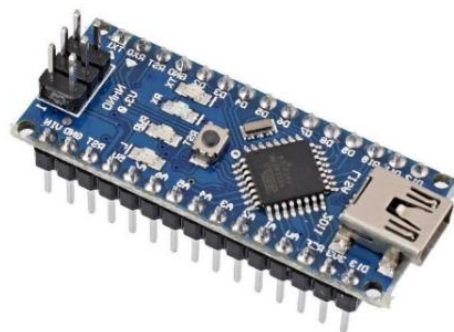
**Microcontrolador:** Usa o ATmega328P, operando a 16 MHz.

**Entradas/Saídas:** 14 pinos digitais (6 PWM) e 8 pinos analógicos.

**Memória:** 32 KB de memória flash, 2 KB de SRAM e 1 KB de EEPROM.

**Conectividade:** Conector USB para programação e alimentação.

**Portabilidade:** Leve e fácil de integrar em protótipos.



*Figura 3 - Arduino Nano*

### 5.1.5 Encoder KY-040

Converte movimentos rotativos (lineares) em pulsos elétricos de onda quadrada.

**Tipo:** Encoder rotativo incremental, que detecta a rotação em duas direções (horário e anti-horário).

**Pinos:** Geralmente possui 5 pinos: VCC, GND, CLK (saída A), DT (saída B) e SW (botão pressionável).

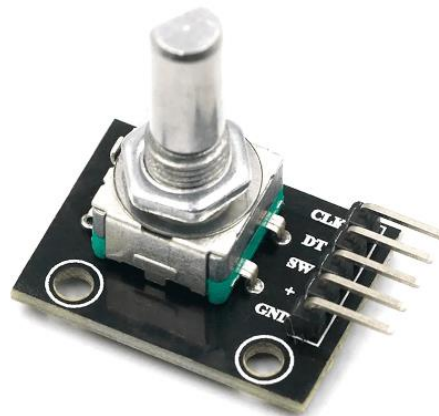
**Resolução:** Alta resolução, tipicamente 20 pulsos por volta, permitindo detecção precisa da posição.

**Botão Integrado:** Inclui um botão que pode ser usado como entrada adicional.

**Fácil de Usar:** Pode ser facilmente integrado com microcontroladores, como Arduino.

**Conexão:** Funciona com tensão de 5V, compatível com a maioria das placas Arduino.

**Aplicações:** Usado em interfaces de usuário, como controle de volume, menus e outros sistemas de navegação.



*Figura 4 - Encoder KY040*

### 5.1.6 Solid State Relay CPC1017N

Baixo RON de aplicação e possui fator automotivo.

**Tipo:** Relé de estado sólido (SSR) com acionamento óptico, ideal para isolamento entre a carga e o controle.

**Tensão de Controle:** Pode ser acionado por uma tensão de controle de 4 a 32V DC.

**Tensão de Carga:** Suporta tensões de carga de até 60V AC ou DC.

**Corrente de Carga:** Capacidade de corrente de até 2A, adequada para aplicações de baixa potência.

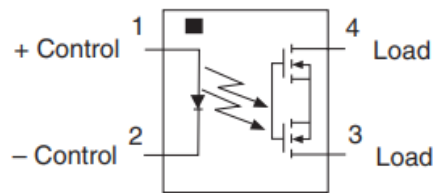
**Isolamento:** Oferece um isolamento de até 4000V entre o lado de controle e o lado de carga.

**Forma de Montagem:** Geralmente disponível em embalagem DIP para montagem em placa de circuito.

**Baixo Ruído:** Opera sem os cliques mecânicos típicos de relés eletromecânicos, resultando em operação silenciosa.

**Durabilidade:** Maior vida útil em comparação com relés eletromecânicos, devido à ausência de partes móveis.

## Pin Configuration



## Switching Characteristics of Normally-Open Devices

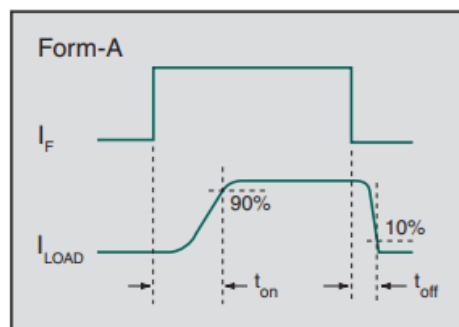


Figura 5 - Solid States

### 5.1.7 CÉLULA DE PEROVSKITA

**Estrutura Cristalina:** Material com uma estrutura cristalina especial.

Importância para Energia Solar:

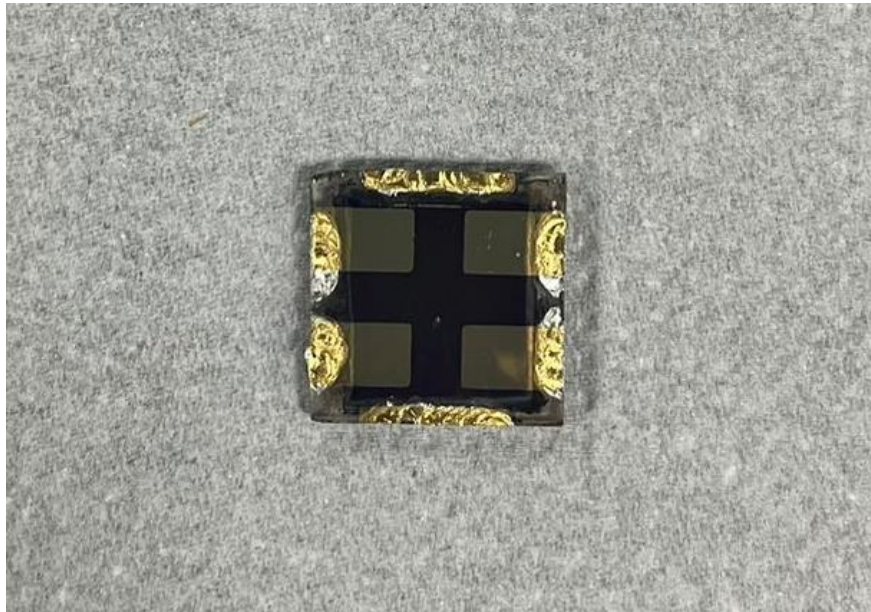
**Eficiência:** Painéis solares altamente eficientes.

**Custo:** Materiais baratos e fáceis de produzir, reduzindo o custo da energia solar.

**Usos Versáteis:** Aplicável em LEDs, lasers e dispositivos eletrônicos flexíveis.

**Flexibilidade:** Pode ser transformada em camadas finas e leves, permitindo uso em diversas superfícies.

**Promissor:** Material que pode levar a inovações em painéis solares e outras tecnologias avançadas.



*Figura 6 - Célula de Perovskita*

### 5.1.8 PLACA DE CIRCUITO IMPRESSO

**Suporte Mecânico:** Base física para fixação e organização dos componentes.

**Conexão Elétrica:** Trilhas de cobre conduzem corrente entre componentes.

**Isolamento Elétrico:** Material isolante evita curtos-circuitos entre trilhas.

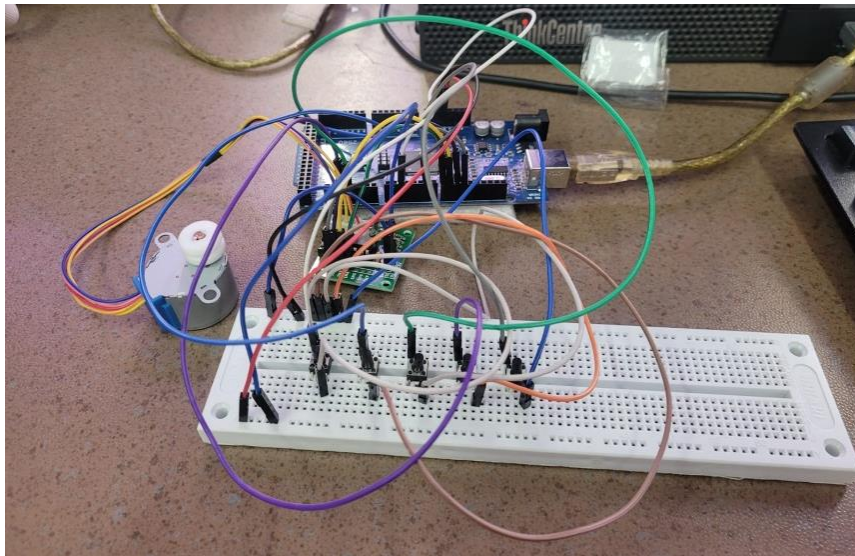
**Compactação do Circuito:** Montagem compacta reduz o tamanho do dispositivo.



*Figura 7 - Placa De PCI*

## 5.2 Processos de Montagem

Após a chegada dos componentes, nos reunimos nas aulas para começar a montagem. Os integrantes que ficaram responsáveis por montar a parte da programação, distribuíram a partir do código os componentes no protoboard para o início dos testes. Paralelamente a isso, a parte mecânica foi sendo desenvolvida com auxílio dos professores de mecânica, e assim foi se estruturando a carcaça do protótipo.



*Figura 8 - Montagem no Protoboard*



*Figura 9 - Início da Montagem*

(Fonte: Os Autores)



*Figura 10 - Placa com Suporte Finalizado*

### **5.3 Princípio de Funcionamento**

Para a execução de sua principal função, que é o giro automático selecionado a cada 90°, a partir do acionamento de botões que coordenaram a sua direção. Foi utilizado, o arduino nano que ao processar as informações do código aciona os botões que possuem a programação específica para um giro de 90° cada um.

Ao acionar o botão o primeiro giro acontece, posicionando a máscara de sombra a 90° graus, e assim por diante, de acordo com o acionamento dos demais botões a máscara se posicionará a cada ângulo reto (90°;180°;270°;360°). Além dos botões destinados ao posicionamento, se encontra o quinto botão que tem a função de alterar o sentido de horário para anti-horário.

Ademais, na sua estação de trabalho será encontra em um suporte à parte, que abrigará os botões em conjunto com o motor de passo que executa a função de manipular a máscara de sombra a partir de seu eixo.

Relacionando ao funcionamento da medição das células: O encoder lê pulsos ao girar seu eixo; para a direita, avança para o próximo contato, e para a esquerda, retrocede para o contato anterior. Um contador armazena o estado atual para gerenciar a seleção de contatos.

Assim os relés (U2, U3, U4 e U5) são acionados a cada pulso do encoder. Ao girar o encoder no sentido horário, o contador aumenta e ativa o relé correspondente ao primeiro contato, seguindo progressivamente até o último. Já os LEDs conectados diretamente ao Arduino indicam qual contato está selecionado. Se o relé correspondente estiver fechado, o LED correspondente acende, permitindo visualizar o contato ativo durante as medições.

- Segue o Esquema Elétrico:

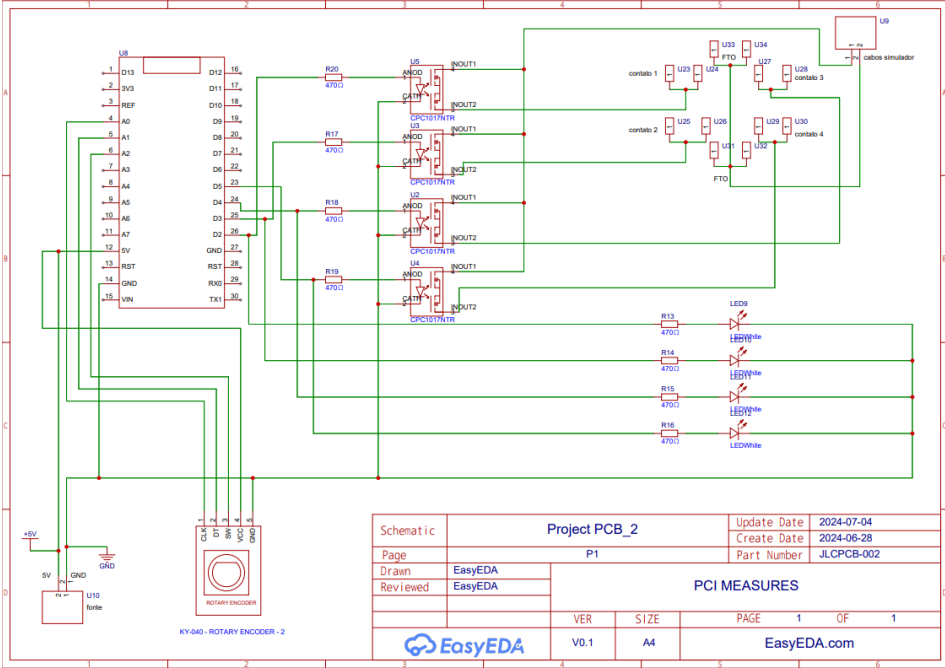


Figura 11 – Esquema elétrico

## 6 PROGRAMAÇÃO

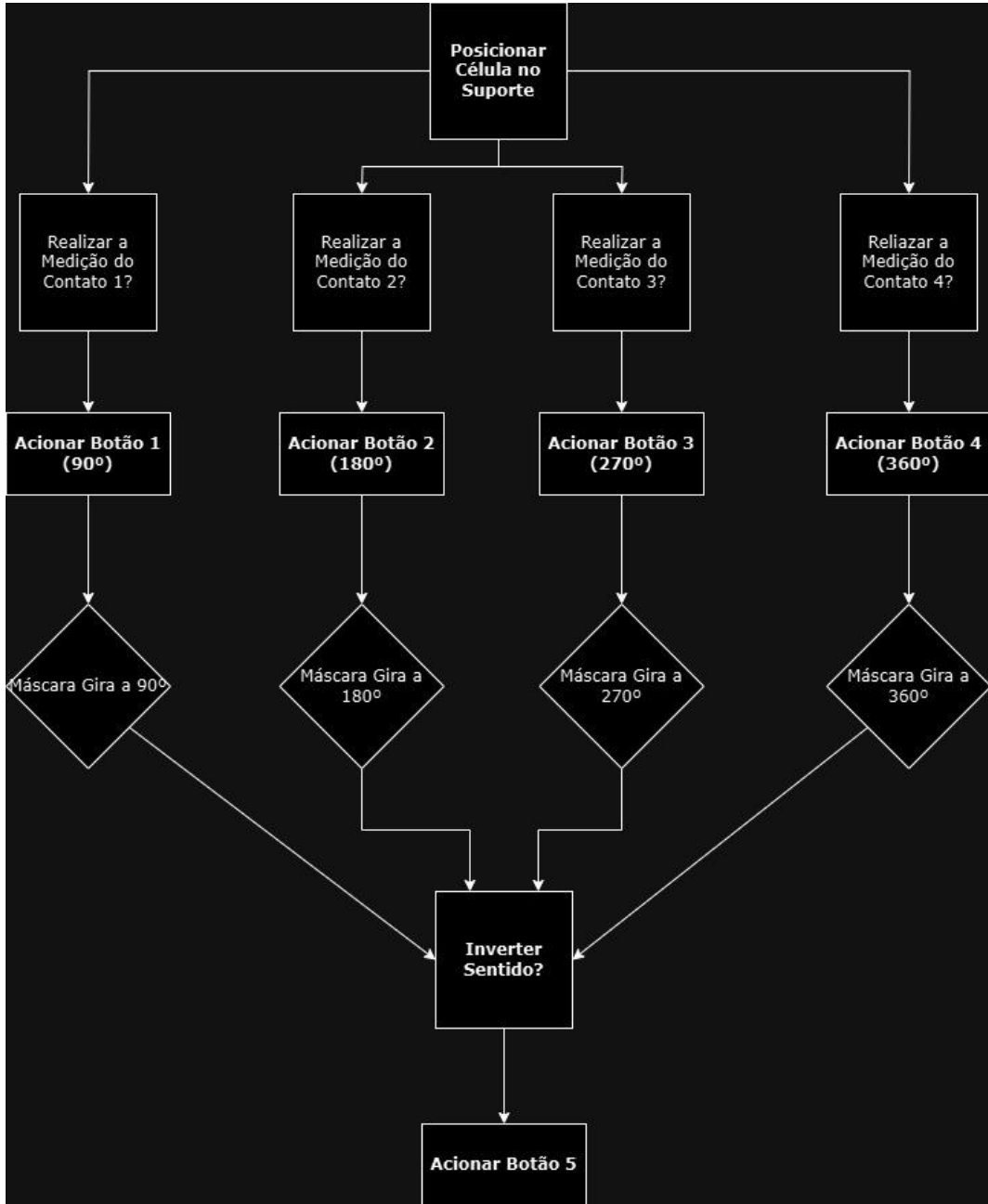


Figura 12 - Programação

## 7 CUSTOS

A variação salarial de um técnico em eletrônica, que pode oscilar entre R\$ 1.800 e R\$ 2.400, reflete diferentes fatores do mercado de trabalho. Essa faixa salarial é influenciada pela localização da empresa, a complexidade das atividades desempenhadas, e a carga horária do estágio. Estágios em grandes centros urbanos ou em empresas de renome tendem a oferecer salários mais altos.

Considerando que a nossa faixa salarial por hora é aproximadamente R\$ 11,20/hora, além disso a equipe técnica varia no horário de trabalho, variando no tempo em que foi dedicado ao protótipo.

A partir disso conseguimos fazer uma média de custo da mão de obra, considerando os tempos dedicados nas aulas, consideramos que o grupo todo trabalhou em conjunto entre 70 a 80 horas.

$$C = 75 \times 11,20 = 840,00 \text{ custo total}$$

Componente	Quantidade	Preço
Filamento Preto PLA (500g)	1	R\$ 75,00
Arduíno Mega	1	R\$ 89,90
Protoboard	1	R\$ 24,99
Kit de Jumpers	2	R\$ 25,98
Botão	4	R\$ 6,00
Módulo do Motor de Passo	1	R\$ 13,99
Motor de Passos	1	R\$ 10,99

*Tabela 4 - Custos dos componentes*

Além de analisar os custos dos componentes, foi necessário o desenvolvimento da tabela de preço da Placa de PCI e da Mascará de Sombra.

<b>Componente</b>	<b>Quantidade</b>	<b>Preço</b>
Arduíno Nano	1	R\$ 29,90
Encoder KY040	1	R\$ 17,20
Resistor 330Ω	4	R\$ 2,00
Resistor SMD 470Ω	20	R\$ 15,99
LED Amarelo	4	R\$ 3,00
Pinos de Contato	12	R\$ 87,90
Conectores	10	R\$ 8,99
Fios Vermelho/Preto	2	R\$ 12,90
Plug Banana	4	R\$ 14,90
SSR CPC1017N	5	R\$ 54,90

*Tabela 5 - Custo Placa de PCI*

Ao concluir os custos do protótipo, foi realizado a criação de uma tabela de preço real do projeto. Uma vez que aborda os gastos de mão de obra, materiais e o quanto seria necessário para desenvolver os componentes essenciais.

<b>Custos</b>	<b>Valores</b>
Custo - Mão de Obra	R\$ 840,00
Custo - PCI	R\$ 247,68
Custo - Máscara de Sombra	R\$ 246,85
<b>Total</b>	<b>R\$ 1.334,53</b>

*Tabela 6 - Custos Gerais*

## 8 RESULTADOS OBTIDOS

Em primeira análise, os objetivos de realizar um suporte automatizado para a medição de células de perovskita, que executasse as funções básicas foi alcançado, uma vez que o protótipo executa de maneira correta o seu papel. Relacionando sua montagem com o tempo, foi concluído no prazo correto, terminado de maneira rápida e sem muitos problemas.

Em segunda análise, o segundo objetivo era relacionar a medição das células com a montagem automatizada seguiu de maneira eficiente, uma vez que a montagem

mecânica se finalizou de maneira simples e eficiente, entrando em funcionamento perfeito com a programação.

Analisando o suporte maior, foi confeccionado para abrigar o suporte das células e os botões destinados ao seu giro. Este suporte teve como objetivo garantir a medição exata e precisa, além de abrigar os componentes de uma forma organizada.

Por fim, o protótipo automatizado foi concluído com sucesso, atingindo seu principal objetivo, garantindo sua funcionalidade e avanço científico.

## 9 CONCLUSÃO

A conclusão do nosso trabalho de conclusão de curso (TCC) destaca a relevância e os avanços alcançados no desenvolvimento de um suporte automatizado para a medição de células solares de perovskita. A energia fotovoltaica, especialmente por meio das células de perovskita, representa uma alternativa viável e promissora em um contexto global que busca fontes de energia mais sustentáveis. Ao longo deste projeto, conseguimos atingir os objetivos propostos, demonstrando a eficácia do protótipo desenvolvido.

Primeiramente, o suporte automatizado foi projetado e construído com sucesso, cumprindo sua função de forma eficiente e dentro do prazo estipulado. A montagem do dispositivo foi realizada de maneira ágil e sem complicações, evidenciando a viabilidade do projeto. Além disso, a integração entre a parte mecânica e a programação permitiu que as medições das células fossem realizadas com precisão, destacando a eficiência do sistema automatizado.

O suporte desenvolvido não apenas facilita a medição das células de perovskita, mas também garante uma disposição organizada dos componentes, essencial para a obtenção de resultados confiáveis. Essa organização é crucial para a validação das medições, contribuindo para o avanço no entendimento e no aproveitamento da energia solar.

Por fim, o protótipo automatizado alcançou seu principal objetivo, consolidando-se como uma ferramenta significativa para a pesquisa e desenvolvimento na área de energia fotovoltaica. Ao proporcionar um método mais acessível e eficiente para a medição das células solares de perovskita, este projeto não apenas contribui para o avanço científico, mas também abre caminho para futuras investigações e melhorias nessa tecnologia inovadora. Acreditamos que este trabalho pode servir como base para o desenvolvimento de novas soluções que potencializem ainda mais o uso da energia solar em um futuro sustentável.

## 10 REFERÊNCIAS

1. Matyas Daboczi; Iain Hamilton; Shengda Xu; Joel Luke; Saurav Limbu; Jinho Lee; Martyn A. McLachlan; Kwanghee Lee; James R. Durrant; Iain D. Baikie; and Ji-Seon Kim. Origin of Open-Circuit Voltage Losses in Perovskite Solar Cells Investigated by Surface Photovoltage Measurement. *CS Appl. Mater. Interfaces* **2019**, *11*, *50*, 46808–46817.
2. Pilipovich, V.A., Sergienya, S.A., Esman, A.K. et al. An Automated System for Measuring the Current-Voltage Characteristics of Solar Cells. *Meas Tech* **48**, 595–600 (2005). <https://doi.org/10.1007/s11018-005-0188-8>.
3. Matthew Wilkins, Richard Beal, Joan E. Haysom, Jeffrey F. Wheeldon, Philippe Mulet, Graeme Jamieson, Nashed Youssef, Dhan Balachandreswaran, Jennifer Fan, Trevor Hall, Stefan Myrskog, and Karin Hinzer "Design of a multiplexer to characterize individual optics at a concentrating photovoltaic test site", *Proc. SPIE 8007, Photonics North 2011, 800725 (8 September 2011)*; <https://doi.org/10.1117/12.906246>.
4. T. L. Companhoni, M. Götz, C. Protasio and I. Müller, "Development of a Wireless Current Measurement Sensor Node for IoT Applications," 2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC), Salvador, Brazil, 2018, pp. 1-7, doi: 10.1109/SBESC.2018.00010.
5. J. W. Wallace, L. C. Diamantides, K. C. Ki and M. W. Butler, "Switched-Antenna Low-Frequency (LF) Radio-Frequency Identification (RFID) for Ornithology," in *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 2, pp. 137-145, June 2020, doi: 10.1109/JRFID.2020.2971534.
6. A. C. Keller, "Relays and Switches," in *Proceedings of the IRE*, vol. 50, no. 5, pp. 932-934, May 1962, doi: 10.1109/JRPROC.1962.288375.
7. Dhakiphale, Amol A., Pratik S. Dhage, and Hrushikesh B. Suryawanshi. "Smart Measuring Tape." *Journal for Research* | Volume 6.2 (2020).
8. Gay, W. (2017). Rotary Encoders. In: Custom Raspberry Pi Interfaces. Apress, Berkeley, CA. [https://doi.org/10.1007/978-1-4842-2406-9\\_8](https://doi.org/10.1007/978-1-4842-2406-9_8).
9. Michael Saliba, Martin Stollerfoht, Christian M. Wolff, Dieter Neher, and Antonio Abate. Measuring Aging Stability of Perovskite Solar Cells. *Joule* **2**, 1019–1027, June 20, 2018.

# 11 ANEXOS

## 11.1 Cronograma de Atividades

	MAIO		JUNHO				JULHO				AGOSTO				SETEMBRO				OUTUBRO			NOVEMBRO						
	23/mai	30/mai	06/jun	13/jun	20/jun	27/jun	04/jul	11/jul	18/jul	25/jul	01/ago	08/ago	15/ago	22/ago	29/ago	05/set	12/set	19/set	26/set	03/out	10/out	17/out	24/out	31/out	07/nov	14/nov	21/nov	
1. RELATÓRIO	RESPONSÁVEIS PELA ETAPA: ANA CAROLINA, ANA CLARA E ISABELLA ELOY																											
1.1. REFERENCIAL TEÓRICO																												
1.1.1. IDENTIFICAR E REVISAR ESTUDOS RELEVANTES, TEORIAS, MODELOS E PESQUISAS RELACIONADAS AO TEMA DO TCC, ESTABELECENDO UMA RELAÇÃO CLARA QUE PERMITA EXPLICAR COMO A LITERATURA EXISTENTE INFORMA E EMBASA A PESQUISA.																												
1.1.2. SINTETIZAR OS CONCEITOS E AS TEORIAS ENCONTRADAS, DESTACANDO AS PRINCIPAIS IDEIAS, DESCOBERTAS E DEBATES NA ÁREA.																												
1.1.3. APRESENTAR OS CONCEITOS FUNDAMENTAIS QUE SERÃO UTILIZADOS NO TRABALHO, CONSTRUINDO UMA BASE TEÓRICA SÓLIDA PARA O DESENVOLVIMENTO DO ESTUDO.																												
1.1.4. JUSTIFICAR A ESCOLHA DOS ESTUDOS E TEORIAS INCLuíDOS NA REVISÃO, EXPLICANDO OS CRITÉRIOS DE SELEÇÃO ADOTADOS;																												
1.1.5. DESENVOLVIMENTO DO RELATÓRIO E INCLUSÃO DOS DADOS OBTIDOS EXPERIMENTALMENTE, CONCLUSÃO																												
1.1.6. FORMATAÇÃO DO TRABALHO ACADÊMICO																												
1.2. METODOLOGIA																												
1.2.1. SIMULAÇÃO DE CIRCUITOS ELETRÔNICOS EM SOFTWARE DE SIMULAÇÃO																												
1.2.2. PROGRAMAÇÃO DE DISPOSITIVOS MICROCONTROLADOS USANDO SOFTWARE DE SIMULAÇÃO																												
1.2.3. USO DO LABORATÓRIO PARA IMPLEMENTAÇÃO E TESTES DE CIRCUITOS ELETRÔNICOS																												
1.2.4. ANOTAR OS DADOS																												
2. FORMALIZAÇÃO E CONSOLIDAÇÃO DO PROJETO EXECUTIVO	RESPONSÁVEIS PELA ETAPA: FELIPE MONTRESOL, GUSTAVO CÉSAR E ISABELA SANTOS																											
2.1. RECURSOS NECESSÁRIOS																												
2.1.1. COMPRA DE MATERIAL E PRAZO DE ESPERA																												
2.2. INÍCIO DA MONTAGEM	RESPONSÁVEIS PELA ETAPA, TODOS OS INTEGRANTES																											
2.2.1. TEMPO NECESSÁRIO PARA MONTAGEM DO PROTÓTIPO																												
2.3. PERÍODO DE TESTE DO PROTÓTIPO	RESPONSÁVEIS PELA ETAPA: ISABELLA ELOY, ANA CLARA E ANA CAROLINA																											
2.3.1. AJUSTES NECESSÁRIOS																												
2.3.2. RESULTADOS OBTIDOS																												
3. APRESENTAÇÃO	RESPONSÁVEIS PELA ETAPA: ISABELA SANTOS, GUSTAVO E FELIPE																											
3.1. CRIAÇÃO DO POWERPOINT DA APRESENTAÇÃO DO TRABALHO																												
3.2. TREINAR PARA A APRESENTAÇÃO																												
4. ENTREGA DO TRABALHO E DEFESA																												

## 11.2 Desenhos

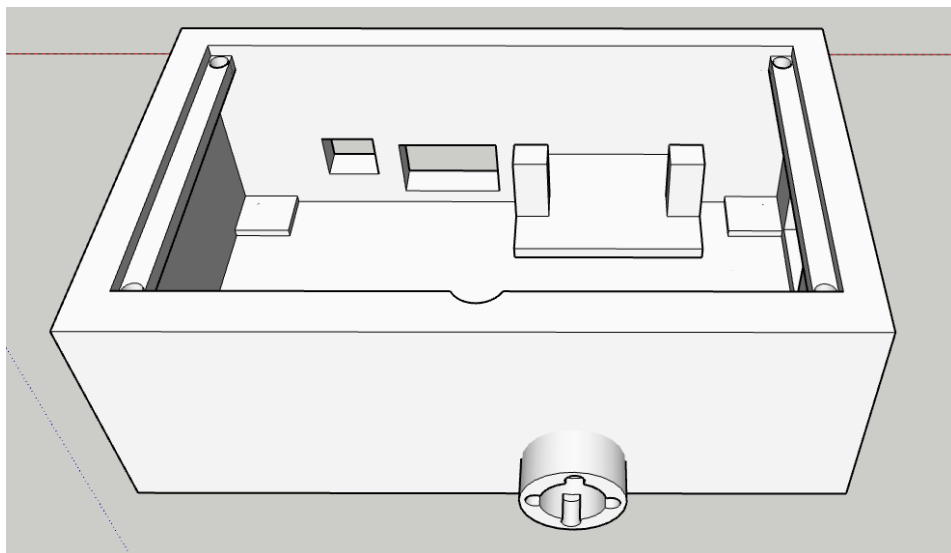
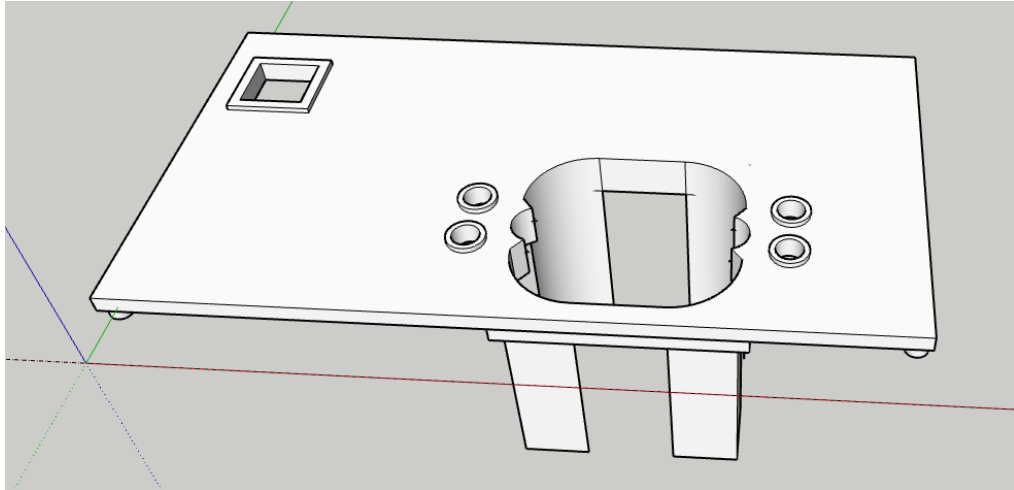
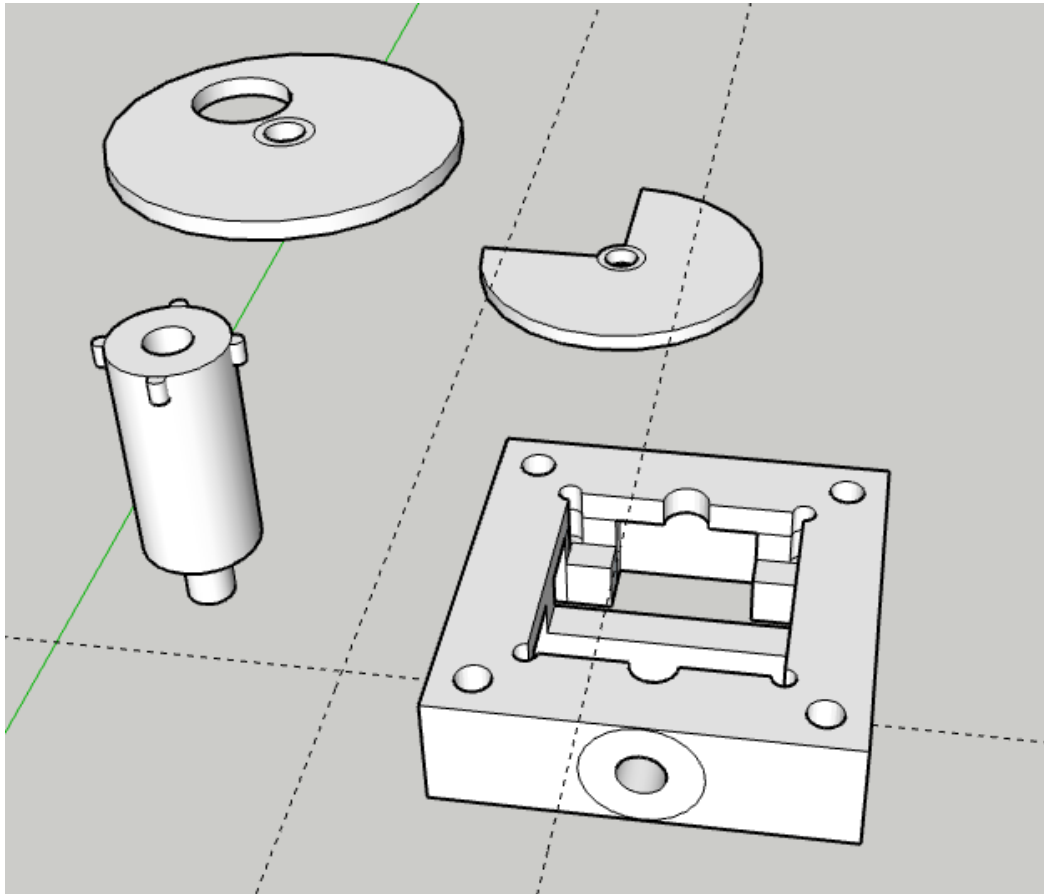


Figura 13- Caixa Final



*Figura 14 - Tampa Da caixa*



*Figura 15 – Suportes Internos da célula*

## 11.3 Programação

### Programação Arduino Mega

```

#define IN1 11
#define IN2 10
#define IN3 9
#define IN4 8

#define BUTTON1 2 // Botão para 90 graus
#define BUTTON2 3 // Botão para 180 graus
#define BUTTON3 18 // Botão para 270 graus
#define BUTTON4 19 // Botão para 360 graus
#define DIR_BUTTON 6 // Botão para controlar direção (horário/anti-horário)

int Steps = 0; // Posição atual do motor
int Direction = 1; // 1: Horário, 0: Anti-horário
int stepsPer90Degrees = 256; // 2048 / 8 (90 graus)

// variável para armazenar o estado acumulado (última posição)
int currentPosition = 0;

void setup() {
  Serial.begin(9600);

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  pinMode(BUTTON1, INPUT); // Sem INPUT_PULLUP
  pinMode(BUTTON2, INPUT); // Sem INPUT_PULLUP
  pinMode(BUTTON3, INPUT); // Sem INPUT_PULLUP
  pinMode(BUTTON4, INPUT); // Sem INPUT_PULLUP
  pinMode(DIR_BUTTON, INPUT); // Sem INPUT_PULLUP
}

void loop() {
  // Verificar o botão de direção
  if (digitalRead(DIR_BUTTON) == HIGH) {
    Direction = 0; // Anti-horário
  } else {
    Direction = 1; // Horário
  }

  if (digitalRead(BUTTON1) == HIGH) {
    moveToPosition(1 * stepsPer90Degrees); // gira 90 graus
    delay(500); // Anti bouncing
  }
  else if (digitalRead(BUTTON2) == HIGH) {
    moveToPosition(2 * stepsPer90Degrees); // gira 180 graus (a partir da posição atual)
  }
}

```

```

    delay(500); // Anti bouncing
}
else if (digitalRead(BUTTON3) == HIGH) {
    moveToPosition(3 * stepsPer90Degrees); // gira 270 graus (a partir da posição atual)
    delay(500); // Anti bouncing
}
else if (digitalRead(BUTTON4) == HIGH) {
    moveToPosition(4 * stepsPer90Degrees); // gira 360 graus (a partir da posição atual)
    delay(500); // Anti bouncing
}
}

// função que movimenta o motor até a nova posição baseada no estado atual
void moveToPosition(int targetPosition) {
    int stepsToMove = targetPosition - currentPosition; // calcula os passos necessários
    a partir da posição atual

    if (stepsToMove < 0 && Direction == 1) {
        Direction = 0; // se for negativo e o motor estiver em horário, inverter a direção
    } else if (stepsToMove > 0 && Direction == 0) {
        Direction = 1; // se for positivo e o motor estiver em anti-horário, inverter a
    direção
    }

    stepper(abs(stepsToMove)); // mover o motor o número de passos necessário
    currentPosition = targetPosition; // atualizar a posição atual
}

void stepper(int nbStep) {
    for (int x = 0; x < nbStep * 8; x++) { // multiplicando por 8 para cada passo completo
        switch(Steps) {
            case 0:
                digitalWrite(IN1, LOW);
                digitalWrite(IN2, LOW);
                digitalWrite(IN3, LOW);
                digitalWrite(IN4, HIGH);
                break;
            case 1:
                digitalWrite(IN1, LOW);
                digitalWrite(IN2, LOW);
                digitalWrite(IN3, HIGH);
                digitalWrite(IN4, HIGH);
                break;
            case 2:
                digitalWrite(IN1, LOW);
                digitalWrite(IN2, LOW);
                digitalWrite(IN3, HIGH);
                digitalWrite(IN4, LOW);
                break;
            case 3:

```

```

    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    break;
case 4:
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    break;
case 5:
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    break;
case 6:
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    break;
case 7:
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    break;
}
delayMicroseconds(1000); // velocidade do motor de passo

// atualiza o passo baseado na direção
if (Direction == 1) {
    Steps++;
} else {
    Steps--;
}

// garantir que o valor de Steps esteja entre 0 e 7
if (Steps > 7) {
    Steps = 0;
}
if (Steps < 0) {
    Steps = 7;
}
}
}

```

## Programação Arduino Nano

```
#define CLK A0
#define DT A1
#define SW A2
#define UM 2
#define DOIS 3
#define TRES 4
#define QUATRO 5

int cont=0;

void saidas(int contador)
{
  switch(contador)
  {
    case(1):{digitalWrite(UM, HIGH);digitalWrite(DOIS,
LOW);digitalWrite(TRES, LOW);digitalWrite(QUATRO, LOW);break;}
    case(2):{digitalWrite(UM, LOW);digitalWrite(DOIS,
HIGH);digitalWrite(TRES, LOW);digitalWrite(QUATRO, LOW);break;}
    case(3):{digitalWrite(UM, LOW);digitalWrite(DOIS,
LOW);digitalWrite(TRES, HIGH);digitalWrite(QUATRO, LOW);break;}
    case(4):{digitalWrite(UM, LOW);digitalWrite(DOIS,
LOW);digitalWrite(TRES, LOW);digitalWrite(QUATRO, HIGH);break;}
    default:{digitalWrite(UM, LOW);digitalWrite(DOIS,
LOW);digitalWrite(TRES, LOW);digitalWrite(QUATRO, LOW);break;}
  }
}

void setup()
{
  pinMode(2, OUTPUT);pinMode(3, OUTPUT);pinMode(4, OUTPUT);pinMode(5,
OUTPUT);
  Serial.begin(9600);
  digitalWrite(UM, LOW);digitalWrite(DOIS, LOW);digitalWrite(TRES,
LOW);digitalWrite(QUATRO, LOW);
}

void loop()
{
  while(digitalRead(DT)==HIGH && digitalRead(CLK)==HIGH){}
```

```
if(digitalRead(DT)==LOW)
{
  while(digitalRead(CLK)==HIGH){}      while(digitalRead(DT)==LOW){}
while(digitalRead(CLK)==LOW){}
  Serial.print("Horário ");
  cont++; if(cont>4){cont=1;} Serial.println(cont); saidas(cont);}

  if(digitalRead(CLK)==LOW)
{
  while(digitalRead(DT)==HIGH){}      while(digitalRead(CLK)==LOW){}
while(digitalRead(DT)==LOW){}
  Serial.print("Anti Horário ");
  cont--; if(cont<1){cont=4;} Serial.println(cont); saidas(cont);}

  delay(10);
}
```