

Manual de confecção

CELTA: Sistema de célula tátil para leitura Braille

Glauber Santiago
Clarissa Bengtson
Douglas Pino
Cristian Pendenza
Juan Santos



Reitoria

Ana Beatriz de Oliveira

Vice-reitoria

Maria de Jesus Dutra dos Reis

Diretor da EDESP

Nassim Chamel Elias

EDESP – Editora de Educação e Acessibilidade da UFSCar

Editores Executivos

Adriana Garcia Gonçalves

Clarissa Bengtson

Douglas Pino

Rosimeire Maria Orlando

Conselho Editorial

Adriana Garcia Gonçalves

Carolina Severino Lopes da Costa

Clarissa Bengtson

Gerusa Ferreira Lourenço

Juliane Ap. de Paula Perez Campos

Marcia Duarte Galvani

Mariana Cristina Pedrino

Nassim Chamel Elias (Presidente)

Rosimeire Maria Orlando

Vanessa Cristina Paulino

Vanessa Regina de Oliveira Martins

Equipe do Projeto

Glauber Santiago

Clarissa Bengtson

Douglas Pino

Cristian Pendenza

Juan Santos

Nadia Perez Pino

Rafaela Marchetti

Meire Moreira Cordeiro

Vanessa Onody

CELTA: Sistema de célula tátil para leitura Braille

Glauber Santiago, Clarissa Bengtson, Douglas Pino, Cristian Pendenza e Juan Santos



Descrição da imagem: A imagem apresenta uma fotografia de um aparelho (o Celta). Tem o formato de um paralelepípedo com 13 cm de frente, 11 cm de lado e 6 cm de altura. A estrutura da caixa é de material plástico de cor branca. Na tampa da caixa, na parte esquerda, abaixo, estão localizados dois botões de apertar e, alinhados ao centro deste botão, existe uma célula braille. Os botões e o corpo da célula são de cor preta. Ao centro da célula constam 6 pequenos pontos de cor branca. A tampa da caixa é fixada por dois parafusos, um na esquerda e outro na direita. Os seguintes dizeres estão inscritos da tampa: 5v, usb e Celta. Eles estão escritos em braille e também entalhados na tampa. Além disso, existem dois traços alinhados, um com o "5v" e o outro com o "usb" e mais duas setas, próximas aos botões, indicando respectivamente para a esquerda e para a direita.

© 2020, dos autores



EDESP-UFSCar

Projeto gráfico e diagramação

Glauber Santiago

Capa

Glauber Santiago e Clarissa Bengtson

Revisão Linguística

Douglas Pino

C394c

CELTA : sistema de célula tátil para leitura Braille / Glauber Santiago ... [et al.]. -- Documento eletrônico -- São Carlos : EDESP-UFSCar, 2022.
194 p.

ISBN: 978-65-89874-31-7

1. Educação especial. 2. Acessibilidade. 3. Braille (Sistema de escrita). 4. Célula tátil. I. Título.

CDD: 371.9 (20^a)

CDU: 371.9

Ficha catalográfica elaborada na Biblioteca Comunitária da UFSCar
Bibliotecário responsável: Ronildo Santos Prado - CRB/8 7325

Todos os direitos desta edição foram reservados aos autores. A reprodução não autorizada desta publicação, no todo ou em parte, constitui violação dos direitos autorais (Lei no 9.610/1998).

Sumário

Agradecimentos	8
1. Introdução	9
1.1 Descrição sumária do Celta	10
1.2 O Celta enquanto uma Tecnologia Assistiva	13
1.2.1 Estado atual das leis sobre acessibilidade no Brasil.....	14
1.3 A <i>cultura maker</i> e sua relação com o projeto.....	17
1.3.1 Modelagem e impressão em 3D	18
1.3.2 Arduino	25
1.4 O display braille.....	34
1.5 A proposta do projeto <i>Sistema de célula tátil para leitura braille</i>	37
1.5.1 Resumo da proposta	37
1.5.2 Público-alvo da proposta.....	38
1.5.3 Problemas e soluções considerados na proposta.....	38
1.6 Histórico do desenvolvimento do projeto	39
1.6.1 Formação da equipe	41
1.6.2 Primeiros protótipos	42
1.6.3 Realização do protótipo final	48
2. Utilização do Celta	51
2.1 Exercícios com o Celta.....	55
2.1.1 Exercícios iniciais com uso do Celta	55
2.1.2 Primeira série (letras de “a” a “j”)	56
2.1.3 Segunda série (letras de “k” a “t”)	58
2.1.4 Terceira série (letras de “u”, “v”, “x”, “y”, “z”, “ç”, “é”, “á” e “ú”)	58
2.1.5 Quarta série (“â”, “ê”, “ô”, “@”, “à”, “í”, “ü”, “õ” e “w”)	59
2.1.6 Quinta série (“, “,” , “.” , “.” , “?” , “!” , “(“ , “)” , aspas, “_”	59

2.1.7 Sexta série (“í”, “ã”, “ó”, sinal de número, apóstrofo, e “-“	60
2.1.8 Sétima série (grifo, letras maiúsculas e \$)	60
2.2 Utilizando o Celta virtual para treinamento de não usuários do braille	60
2.3 Ideias para uso com jogos educacionais	62
2.3.1 Jogo descubra o caractere	62
2.3.2 Descubra a palavra	62
3. Como construir um Celta	63
3.1 Apresentação	63
3.1.1 Quais conhecimentos são necessários para a construção?66	
3.1.2 Qual pessoal é necessário?	66
3.1.3 Que infraestrutura, equipamentos e insumos são necessários?	66
3.1.4 Quais os softwares e sites que serão utilizados?	69
3.2 Confecção da célula	69
3.2.1 Projeto CAD da célula	70
3.2.2 Configurações do fatiador para a célula	76
3.2.3 Impressão da célula	84
3.3 Confecção do box	93
3.3.1 Projeto CAD do box	94
3.3.2 Configurações do fatiador e de impressão para o box	98
3.4 Confecção da placa de controle	104
3.4.1 Componentes eletrônicos necessários	105
3.4.2 Projeto e diagrama elétrico	106
3.4.3 Montando uma placa ilhada	114
3.4.4 Confeccionando uma placa específica para o Celta	131
3.5 Preparação dos Motores	145
3.6 Montagem final	152

3.7 Resumo sobre a confecção e a montagem.....	166
4. Software do Celta: instalação e utilização.....	167
4.1 Carregamento no Arduino	168
4.2 Software leitor para o computador	171
4.3 Detalhamento sobre o software do Arduino para desenvolvedores	171
4.4 Instalação especial para desenvolvedores	180
5. Aprimoramentos e desdobramentos futuros para o projeto .	182
6. Referências	185
6.1 Referências de estudo	185
6.2 Referências utilizadas no texto	186
Apêndice: Repositório de Arquivos	188
Baixe aqui os arquivos disponibilizados para a realização do projeto	188
Links para material do projeto em sites externos.....	189
Links para os vídeos do projeto em alta definição	190

Agradecimentos

Em primeiro lugar, gostaríamos de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), que financiou o projeto por meio do Edital nº 3/2018 - Ferramentas de acessibilidade. Agradecemos ainda às equipes de Desenvolvimento, de TI, de Acessibilidade e de Editoração, da Secretaria Geral de Educação a Distância da UFSCar. Sem o apoio dos membros destas equipes, este projeto não poderia ser concebido ou realizado. Este agradecimento se estende à SEaD como um todo, representada pela professora Marilde Santos, por incentivar e amparar esta iniciativa. Somos gratos também ao Curso de Especialização em Educação e Tecnologias da UFSCar, pelo apoio que forneceu em momentos difíceis do projeto, e ainda à Fundação de Apoio Institucional ao Desenvolvimento Científico (FAI), à Pró-reitoria de Extensão (ProEx), à Pró-Reitoria de Administração (ProAd), ao Departamento de Artes e Comunicação (DAC) e à Secretaria de Ações Afirmativas, Diversidade e Equidade (SAADE) da UFSCar. Por fim, somos gratos ao colaborador Luiz Flávio Bravo pelo apoio que forneceu à SEaD durante os anos que esteve conosco. Sem ele dificilmente haveríamos idealizado este empreendimento.

1. Introdução

Este manual (disponibilizado no versão e-book e na versão site: <<https://celta.ufscar.br>>) objetiva trazer informações para a difusão e utilização do resultado de um projeto intitulado *Sistema de célula tátil para leitura braille*, elaborado pela equipe da Secretaria Geral de Educação a Distância (SEaD) da Universidade Federal de São Carlos (UFSCar). O projeto obteve apoio financeiro da *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES), por meio do edital nº 3/2018 - *Ferramentas de acessibilidade*.

Como resultado do projeto, tem-se os seguintes elementos:

- Desenvolvimento de um hardware e software para a leitura braille (display braille), nomeados por nós como *Celta (Célula tátil)*.
- Disponibilização de modo livre de toda a documentação necessária para a replicação do hardware e do software.
- Elaboração de manuais (site, e-book, vídeos etc.) com instruções para a replicação.
- Elaboração de material de divulgação, contextualização e esclarecimento sobre a utilização e aplicabilidade do Celta.

No presente manual, apresentaremos esses elementos para possibilitar ao interessado eficiência na busca por. Sendo assim:

Na *Introdução*, temos a descrição sumária do Celta e uma contextualização sobre acessibilidade, incluindo o estado atual das leis sobre acessibilidade no Brasil. Temos, ainda, uma seção sobre a *cultura maker* e sua relação com o projeto, na qual explicamos um pouco sobre modelagem e impressão em 3D, Arduino e alguns rudimentos de eletrônica, importantes para o leitor que não está familiarizado com este tipo de tecnologia. Na sequência, tratamos de questões relativas ao braille e display braille. Finalizamos a *Introdução* com uma parte voltada à proposta de projeto que foi

realizada, incluindo o resumo da proposta, o público-alvo, os problemas e as soluções consideradas e o histórico do desenvolvimento do projeto (formação da equipe, primeiros protótipos e realização do protótipo final).

No Capítulo 1, *Utilização do Celta*, são apresentadas as formas de uso do sistema, incluindo ideias para uso educacional e propostas de cursos de capacitação.

O Capítulo 2, *Como construir um Celta*, é, de forma objetiva, o manual de construção para aqueles que desejarem replicar o sistema. Conta com uma breve apresentação e, na sequência, descreve: conhecimentos necessários, profissionais envolvidos, infraestrutura, equipamentos, insumos, softwares e componentes para a construção. Depois, o Capítulo prossegue para: as instruções de confecção da célula, do box e da placa de controle; e sobre a preparação dos motores e a montagem final.

No Capítulo 3, *Instruções sobre a instalação e utilização do Software do Celta*, são abordados os temas relativos ao software desenvolvido para a utilização do hardware do sistema Celta.

Por fim, no Capítulo 4, *Aprimoramentos e desdobramentos futuros para o projeto*, deixamos alguns apontamentos para futuras melhorias no sistema. Há, ainda, na seção *Referências*, os textos com os quais trabalhamos para executar o projeto, bem como as com referências utilizadas neste texto.

1.1 Descrição sumária do Celta

O Celta é um produto, um sistema computacional composto de hardware (máquina) e software (programa), que funciona como um periférico de computador¹. Trata-se de um display braille com uma única célula e dois botões de controle. Em sua utilização normal, o Celta é conectado ao computador por meio de um cabo USB. Com o software de interface aberto, o usuário escolhe um texto qualquer em seu computador e o copia para a área de transferência do computador (comandos de copiar, Ctrl+c, e colar,

¹ Como um monitor, uma impressora etc.

Ctrl+v). Então, o texto colado na interface pode ser lido, sob forma de caracteres braille, com a ponta dos dedos, no hardware. Esta leitura se dá com um caractere de cada vez. O usuário utiliza dois botões, para avançar ou retroceder, na apresentação dos caracteres. Vide a Figura 1.1 para entender melhor sobre a Celta.



Figura 1.1 – Sistema Celta pronto para ser utilizado.

O Celta foi desenvolvido para ser replicado por instituições, entidades ou indivíduos interessados em garantir às pessoas com deficiência visual, que leem em braille, acesso a conteúdos textuais. Por isso, disponibiliza todas as informações de forma gratuita e ainda está baseado em tecnologias que baixo custo e de razoável facilidade de utilização. Estimamos que com menos de R\$

300,00 cada unidade possa ser elaborada, em termos de materiais utilizados.

Em relação à utilização, vislumbramos que o Celta facilite atividades de letramento em braille para a divulgação deste conhecimento para o treinamento, nesta leitura, de uma maior quantidade de pessoas. Também cremos que, por se tratar de um projeto aberto possa agregar no fomento do desenvolvimento tecnológico para a acessibilidade, no Brasil.

O Quadro 1.1 apresenta um esquema geral sobre o Celta. O detalhamento destes aspectos será realizado no decorrer dos Capítulos 1, 2, 3 e 4 deste livro.

Quadro 1.1 – Esquema com uma visão geral sobre o Celta.

Objetivo	<ul style="list-style-type: none"> ○ O objetivo principal do Celta é desenvolver um display Braille, composto por uma ou mais células táteis, de: <ol style="list-style-type: none"> 1. código aberto 2. baixo custo □ O dispositivo é indicado para pessoas com deficiência visual ou surdocegueira.
Proposta	<p>Hardware</p> <ul style="list-style-type: none"> ○ Célula Tátil impressa em 3D, acionada por seis micromotores DC 5V. ○ Módulo eletrônico para controle e inversão dos motores de acionamento da célula, utilizando 3 CIs (circuitos integrados), do tipo L293D (circuito ponte-h). <p>Software</p> <ul style="list-style-type: none"> ○ Desenvolvido em C++ com interface gráfica e para o controle do Arduino.
Funcionamento	<ul style="list-style-type: none"> ○ Cada par de motores é controlado por meio de um L293D, que realiza o papel de uma Ponte H – um circuito especial que permite realizar a inversão da direção (polaridade) da corrente. ○ O Arduino envia sinais para esse controlador que irá alternar o sentido de giro dos motores. ○ Cada conjunto de instruções compõe um estado da célula que atualiza o display com o caractere enviado. ○ O equipamento disponibiliza dois botões para que o usuário navegue através do texto exibido pelo software, dessa forma um botão é usado para avançar caracteres enquanto o outro permite retroceder com o

	caractere.
Estados dos motores	<ul style="list-style-type: none"> ○ Cada um dos seis motores possui três estados: 1. para a esquerda: usado para que o pino fique ativado em alto relevo; 2. para a direita: usado para que o pino fique desativado e escondido; 3. inativo: quando o motor está desligado após haver mudado de estado.
Arduino e CELTA	<ul style="list-style-type: none"> ○ Arduino é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador + uma linguagem de programação padrão. ○ Está diretamente ligado à nossa Célula Tátil, ativando ou desativando cada um dos seus pontos. ○ Basicamente, o programa fica num loop para verificar se existe algum comando a ser executado. ○ As ações podem surgir dos botões (avançar/retroceder) ou serem recebidas via comunicação serial do software.
Interface gráfica	<ul style="list-style-type: none"> ○ Através do framework QT Creator, o projeto conta com uma interface gráfica na linguagem C++. ○ Esta interface amigável exibe o conteúdo num bloco de texto, letra a letra, em braille, no dispositivo Celta. ○ O programa percorre cada letra do conteúdo e envia o respectivo código da letra com os pontos que devem estar ativados ou desativados para que a letra esteja representada no formato braille. ○ Por exemplo: se enviar o comando no formato "L1L2L3D4L5L6", todos os pontos exceto o quatro serão ativados.
Navegação entre caracteres	<ul style="list-style-type: none"> ○ Para a navegação, temos as interações com os botões: 1. clique rápido é para avançar/retroceder um caractere 2. clique longo é para pular/voltar uma palavra inteira 3. clique de ambos os botões de forma breve faz com que o cursor da letra atual volte ao começo

1.2 O Celta enquanto uma Tecnologia Assistiva

O Celta foi concebido em um contexto no qual as necessidades específicas de cada pessoa são consideradas na e pela sociedade. Ou seja, os conhecimentos e todos os desenvolvimentos tecnológicos devem considerar aspectos de

acessibilidade para uma variedade de públicos, inclusive as chamadas *peças com deficiência* (PcD). Conforme a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência), nº 13.146, de julho de 2015, considera-se acessibilidade a:

possibilidade e condição de alcance para utilização, com segurança e autonomia, de espaços, mobiliários, equipamentos urbanos, edificações, transportes, informação e comunicação, inclusive seus sistemas e tecnologias, bem como de outros serviços e instalações abertos ao público, de uso público ou privados de uso coletivo, tanto na zona urbana como na rural, por pessoa com deficiência ou com mobilidade reduzida.

Enquanto Tecnologia Assistiva (TA), o Celta apresenta todas as características para figurar como tal, uma vez que encaixa-se na definição de TA fornecida pela referida Lei:

tecnologia assistiva ou ajuda técnica: produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social (LEI nº 13.145/2015).

1.2.1 Estado atual das leis sobre acessibilidade no Brasil

Em se tratando de legislação brasileira a respeito da acessibilidade, temos como marco a Lei 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas com deficiência ou mobilidade reduzida. Esta lei vai ao encontro da *Declaração de Salamanca*, documento concebido na Conferência Mundial sobre Educação Especial, em 1994, em que se buscou a “melhoria do acesso à educação para a maioria daqueles cujas necessidades especiais ainda se encontram desprovidas” (p. 1).

O Decreto Federal nº 5.296, de 2 de dezembro de 2004, que regula a referida Lei, apresenta em seu Artigo 8º a Acessibilidade como “condição para utilização, com segurança e autonomia, total ou assistida, dos espaços, mobiliários e equipamentos urbanos, das

edificações, dos serviços de transporte e dos dispositivos, sistemas e meios de comunicação e informação”.

No mesmo Decreto, Artigo 5º, temos a classificação das deficiências (vide o quadro 1.2):

Quadro 1.2 – Classificação das deficiências segundo do Art 5º do Decreto Federal nº 5.296/2004

Deficiência Física	Alteração completa ou parcial de um ou mais segmentos do corpo humano, acarretando o comprometimento da função física, apresentando-se sob a forma de paraplegia, paraparesia, monoplegia, monoparesia, tetraplegia, tetraparesia, triplegia, triparesia, hemiplegia, hemiparesia, ostomia, amputação ou ausência de membro, paralisia cerebral, nanismo, membros com deformidade congênita ou adquirida, exceto as deformidades estéticas e as que não produzam dificuldades para o desempenho de funções.
Deficiência Auditiva	Perda bilateral, parcial ou total, de quarenta e um decibéis (dB) ou mais, aferida por audiograma nas frequências de 500Hz, 1.000Hz, 2.000Hz e 3.000Hz.
Deficiência Visual	Cegueira, na qual a acuidade visual é igual ou menor que 0,05 no melhor olho, com a melhor correção óptica; a baixa visão, que significa acuidade visual entre 0,3 e 0,05 no melhor olho, com a melhor correção óptica; os casos nos quais a somatória da medida do campo visual em ambos os olhos for igual ou menor que 60o; ou a ocorrência simultânea de quaisquer das condições anteriores.
Deficiência Mental	Funcionamento intelectual significativamente inferior à média, com manifestação antes dos dezoito anos e limitações associadas a duas ou mais áreas de habilidades adaptativas, tais como: 1. comunicação; 2. cuidado pessoal; 3. habilidades sociais; 4. utilização dos recursos da comunidade; 5. saúde e segurança; 6. habilidades acadêmicas; 7. lazer; e 8. Trabalho.
Deficiência Múltipla	Associação de duas ou mais deficiências.

Sendo uma forma de atualização e complemento, foi promulgada a Lei nº 13.146, de 06 de julho de 2015 – Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência), cuja definição de pessoa com deficiência é a seguinte: “aquela que tem impedimento de longo prazo de natureza física, mental, intelectual ou sensorial, o qual, em interação com uma ou mais barreiras, pode obstruir sua participação plena e efetiva na sociedade em igualdade de condições com as demais pessoas”.

O Celta foi desenvolvido assentado nos artigos 72, 77 (parágrafos 1º e 3º) e 78 (parágrafo único) desse Estatuto, que tratam especificamente da importância de se criar tecnologia assistiva:

Art. 72. Os programas, as linhas de pesquisa e os projetos a serem desenvolvidos com o apoio de agências de financiamento e de órgãos e entidades integrantes da administração pública que atuem no auxílio à pesquisa devem contemplar temas voltados à tecnologia assistiva.

Art. 77. O poder público deve fomentar o desenvolvimento científico, a pesquisa e a inovação e a capacitação tecnológicas, voltados à melhoria da qualidade de vida e ao trabalho da pessoa com deficiência e sua inclusão social.

§ 1º O fomento pelo poder público deve priorizar a geração de conhecimentos e técnicas que visem à prevenção e ao tratamento de deficiências e ao desenvolvimento de tecnologias assistiva e social.

§ 3º Deve ser fomentada a capacitação tecnológica de instituições públicas e privadas para o desenvolvimento de tecnologias assistiva e social que sejam voltadas para melhoria da funcionalidade e da participação social da pessoa com deficiência.

Art. 78. Devem ser estimulados a pesquisa, o desenvolvimento, a inovação e a difusão de tecnologias voltadas para ampliar o acesso da pessoa com deficiência às tecnologias da informação e comunicação e às tecnologias sociais.

Parágrafo único. Serão estimulados, em especial:

I - o emprego de tecnologias da informação e comunicação como instrumento de superação de limitações funcionais e de barreiras à comunicação, à informação, à educação e ao entretenimento da pessoa com deficiência.

1.3 A cultura maker e sua relação com o projeto²

Conforme o item anterior, vimos que existe uma demanda social e legal que fomentou a necessidade do Celta. Porém, não se pode descartar um outro aspecto do mundo hodierno que foi fundamental para esta incursão, trata-se da *cultura*, ou *movimento maker*.

O *movimento maker* está diretamente relacionado à ideia do *faça você mesmo* ou *DIY (do it yourself)* e tem como princípios o: fazer, compartilhar, dividir, dar, aprender, equipar, brincar, participar, apoiar e mudar (HATCH, 2013). Por meio dos avanços tecnológicos, as pessoas conseguem criar artefatos com facilidade e de baixo custo, viabilizando e fomentando ideias criativas (SANTIAGO; OLIVEIRA, 2019). Para o desenvolvimento das atividades *maker*, alguns laboratórios (FabLabs) ou oficinas foram criados para facilitar o uso dos materiais e integrar as comunidades.

Sendo assim,

O uso de materiais diversos nos equipamentos de fabricação 3D combina muito com a filosofia da cultura maker, porque ela não só valoriza a capacidade de criação e de realização do indivíduo e o compartilhamento de conhecimentos, como foca o uso de materiais alternativos muitas vezes ignorados pela indústria (VISHBOX, 2019, p. 1).

O *movimento maker*, juntamente ao uso da impressora 3D, estimula a elaboração de diversos objetos que incentivam o aluno a trabalhar e aprender com conceitos de marcenaria, mecânica, costura, programação, robótica, design etc. (VISHBOX, 2019). Além

² Obs.: Partes do texto deste item já figuram em outras publicações dos autores relacionadas com este tema. A saber: SANTIAGO, G. L. A.; OLIVEIRA, C., D. de. **Introdução à modelagem e impressão**. São Carlos: Edutec (curso de especialização em Educação e tecnologias), 2020, 35 p.

disso, desenvolve a aprendizagem criativa, que pode ser entendida “como a transformação pessoal a partir da conquista de novas habilidades e conhecimentos. Esse processo ocorre através do engajamento direto na realização de projetos particulares ou coletivos que sejam genuinamente relevantes para os envolvidos” (SKORTZARU, 2017, s/p.).

Durante a concepção deste projeto, a equipe da SEaD estava envolvida diretamente com esta cultura maker, inclusive elaborando modelagens em 3D para aplicações em acessibilidade. Esta incursão foi fundamental para que nosso olhar fosse voltado para a possibilidade de construção de um display braille acessível, também financeiramente.

A seguir, vamos introduzir o leitor em uma das aplicações mais representativas do *movimento maker*, a impressão em 3D.

1.3.1 Modelagem e impressão em 3D3

O mundo físico é modificado por animais e humanos há milhões de anos, conforme a arqueologia. Um macaco prego, para quebrar uma castanha, tem a destreza tecnológica de colocá-la sobre uma pedra maior e golpeá-la com uma pedra menor, utilizando-a como martelo⁴. Esse ato de seleção de um artefato natural lhe dá uma boa vantagem de sobrevivência. Um hominídeo, por sua vez, utiliza uma tecnologia a qual podemos pensar que é ancestral à impressão em 3D, isto é, a construção de objetos com sílex. De fato, veremos que parte do processo de modelagem parece bastante pré-histórico, quando destruimos coisas para dar forma a elas.

³ Obs.: Partes do texto deste item já figuram em outras publicações dos autores relacionadas com este tema. A saber: SANTIAGO, G. L. A.; BENGTON, C.; PINO, D.; PENDENZA, C; KAWAKAMI, C. **Célula tátil Braille**: Desenvolvimento de uma Ferramenta da Acessibilidade. In: Anais do ESUD/CIESUD, 2019.

⁴ Vide a seguinte reportagem para mais detalhes:

<https://revistapesquisa.fapesp.br/2019/06/24/cultura-de-uso-de-ferramentas-por-macacos-prego-variou-ao-longo-de-3-mil-anos/>.

Para criarmos objetos físicos, podemos pensar em dois princípios: adição e subtração. Por exemplo, uma estátua grega em bronze era feita a partir da adição do metal derretido por meio de um molde. Já uma cópia romana dessa estátua em mármore era obtida a partir da subtração de material. Na produção de objetos por impressoras 3D esses princípios perduram. Porém, a tecnologia tem focado na **adição**.

Então, vamos para o início da década de 1980 com Chuck Hull, que foi o pioneiro na moderna tecnologia de impressão em 3D. A Figura 1.2 apresenta a primeira peça gerada com uma impressora 3D. Esse modelo de impressora utilizava o processo da estereolitografia, que consiste em ir aquecendo com um laser elementos líquidos que são solidificados, dando forma ao objeto.



Figura 1.2 – A primeira peça impressa em impressora 3D. Fonte: Hull (2015, p. 26).

Segundo Santiago e Oliveira,

Em uma linguagem simples, pode-se dizer que a impressão 3D (também conhecida como prototipagem rápida) consiste basicamente em se ir “desenhando” ou construindo, fragmento a fragmento, um objeto. Dentre as várias tecnologias possíveis para a formação dos objetos em 3D, uma das mais econômicas e utilizadas são os filamentos de elemento plástico, que é derretido e injetado em uma posição específica na impressora (SANTIAGO; OLIVEIRA, 2019, p. 285).

Esse processo assemelha-se a uma pistola de cola quente, em que existe um bastão plástico que é aquecido e injetado por um pequeno orifício. A diferença é que na impressora 3D esse plástico é um filamento (semelhante a uma linha de pesca grossa) e todo o processo é guiado por motores controlados computacionalmente. A Figura 1.3 ilustra uma impressora 3D alimentada com um filamento de cor alaranjada.



Figura 1.3 – Impressora 3D com filamento. Fonte: Dos autores.

Segundo Campbell (2011), a criação de peças em 3D possui duas fases. Uma é a modelagem em software, e a outra é a impressão da peça conforme a modelagem realizada. Para a primeira etapa, temos uma variedade de softwares de modelagem, incluindo o *Paint 3D* (já incluso no sistema operacional *Windows®*), o *Blender 3D*, o *3D Studio Max* e o site *AutoDesk® Tinkercad*. Para o iniciante, este último é o ideal, por ser gratuito e de uso intuitivo (vide a Figura 1.4). “A ideia principal deste tipo de modelagem é de adição e modificação de formas tridimensionais positivas (que

geram conteúdo no mundo físico) e negativas (que não geram ou, melhor, “destroem” conteúdo do mundo físico)” (SANTIAGO; OLIVEIRA, 2019, p. 285).

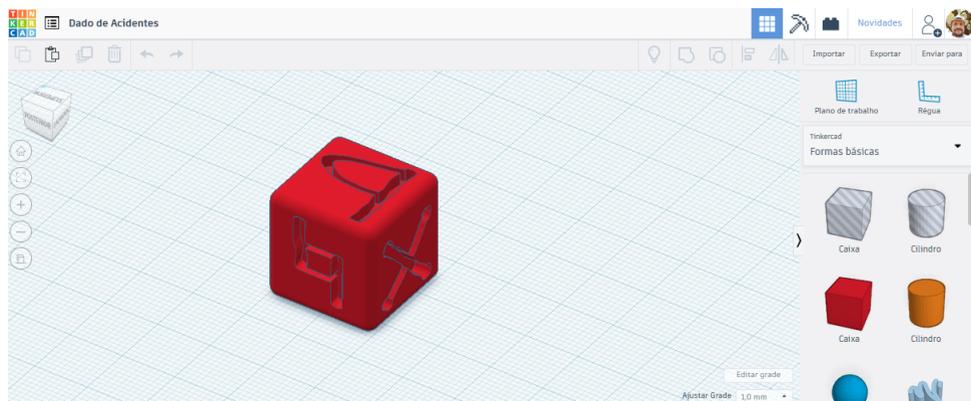


Figura 1.4 – Visão geral do software on-line Tinkercad com um dado musical sendo modelado. Fonte: Dos autores.

No Capítulo 3, quando falarmos da confecção das partes do Celta, iremos utilizar *Tinkercad*. Por isso, é importante atentar para a presente descrição. O *Autodesk Tinkercad* é uma iniciativa da empresa *Autodesk* para incentivar o *movimento maker* como um todo. Busca ser um ponto de convergência entre ideias, criadores, educadores e estudantes. No site, há diversas áreas, incluindo *webinars para educadores*, *Conectando-se com os pais*, *Classes virtuais*, *Galeria*, *Blog*, *Aprenda*, *Ensinar (salas de aula)*, além da aplicação de desenvolvimento de projetos em 3D, circuitos e blocos de código. Vale muito a pena explorar todos esses recursos, mas aqui no Celta vamos focar apenas em projeto 3D.

Primeiros passos no Tinkercad

Para utilizar o *Autodesk Tinkercad*, inicie criando uma conta! Siga os passos:

- Entre no site: <https://www.tinkercad.com>.
- Clique no botão INSCREVER-SE para criar uma conta.

- Clique na opção Criar uma conta pessoal, acrescente seus dados e aceite as opções necessárias.

Depois, entre na página inicial do *Tinkercad*. A dica para ir para a página inicial é clicar no ícone que aparece à esquerda superior da tela, conforme ilustra a Figura 1.5.

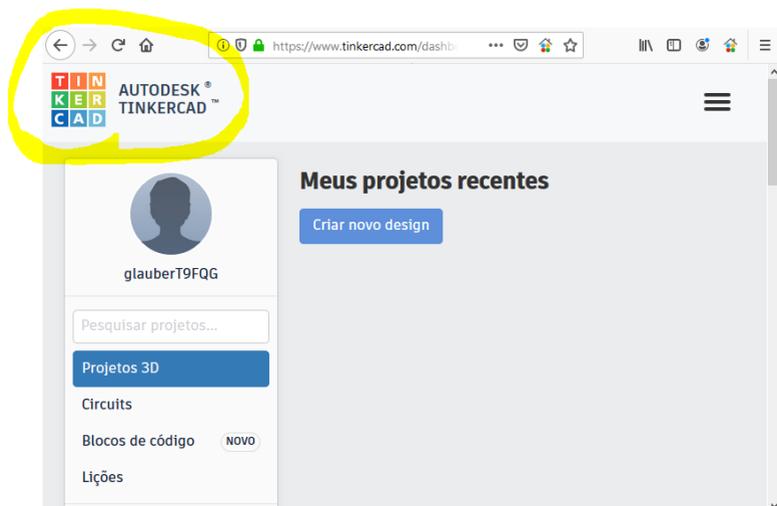


Figura 1.5 – Localização do ícone para ir para a tela inicial.

Na tela inicial, clique em *Criar novo design*. Então será apresentada a tela principal de trabalho para os projetos em 3D. Olhe com atenção a Figura 1.6 para conhecer os principais elementos.

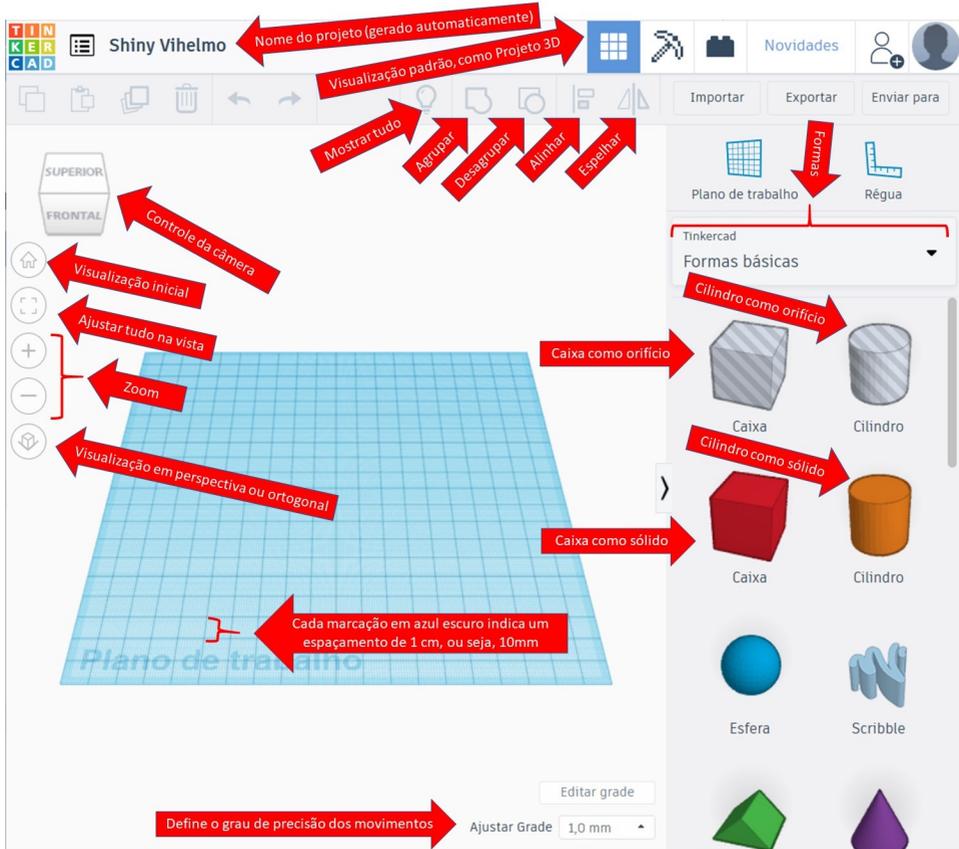


Figura 1.6 – Área de trabalho para projetos em 3D.

Para os mais inexperientes em modelagem 3D, os itens mais importantes a serem considerados são os seguintes: *controle da câmera*, *visualização em perspectiva* ou *ortogonal* e a ideia de *sólidos* e *orifícios*. Cada um desses elementos deve ser cuidadosamente entendido e praticado para que o aprendiz não fique frustrado em seu caminho na modelagem.

1.3.2 Arduino⁵

Assim como a impressão 3D é útil para a prototipagem rápida de objetos, na *cultura maker* o Arduino é uma tecnologia utilizada para a prototipagem envolvendo dispositivos eletrônicos e computacionais. Em nosso Celta utilizamos tal tecnologia. Por isso, é importante que o interessado entenda um pouco sobre esse tema, antes de adentrar na confecção do Celta.

Arduino é uma tecnologia desenvolvida para a prototipagem rápida para a física computacional. Ou seja, uma forma de criação veloz de aplicações que envolvem computação em interação com ambientes físicos, reais. A tecnologia do Arduino envolve três aspectos a serem considerados: um de hardware (placas com componentes eletrônicos), um de software (plataforma para elaboração de programas, bibliotecas etc.) e um legal (possibilidade de utilizar a tecnologia sem muita necessidade de preocupações com direitos autorias).

Placas

Em relação ao hardware, existem diversos modelos de placas, conforme mostra a Figura 1.7. Nela, da esquerda para a direita, temos os seguintes modelos: Lilypod, Nano, Uno e Mega.

⁵ Obs.: Partes do texto deste item já figuram em outras publicações dos autores relacionadas com este tema. A saber: SANTIAGO, Glauber Lúcio Alves. **Arduino para o ensino musical**. Ebook da disciplina Recursos tecnológicos para o ensino de música, Departamento de Artes e Comunicação, UFSCar, 2020, 65. p.

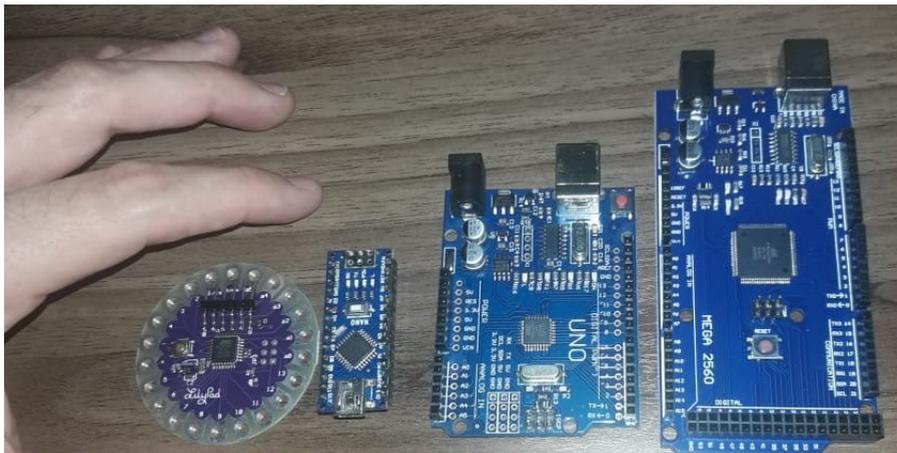


Figura 1.7 – Arduinos LilyPod, Nano, Uno e Mega. Fonte: Dos autores.

Vamos focar, inicialmente, no hardware utilizando como exemplo uma das placas que existem com a tecnologia chamada de *Arduino Uno*. A Figura 1.8 apresenta um *Arduino Uno*. Ao centro, temos o processador que é, de fato, o cérebro da placa. Os demais componentes são para alimentação, comunicação e outras questões secundárias.



Figura 1.8 – Arduino Uno. Fonte: Dos autores.

Em uma linguagem mais simples, essa placa é um computador, só que no lugar de mouse, teclado e monitor são ligados fios (cabos de eletricidade). Na Figura 1.8, temos conexões para ligar cabos. Verifique o nome das conexões: Na esquerda vemos, de cima para baixo, as anotações: *5V*, *RES*, *3.5V*, *GND*, *GND*, *Vin*, *A0*, *A1*, *A2*, *A3*, *A4* e *A5*. Na direita, vemos *SCL*, *DAS*, *AREF*, *GND* e depois de 13 até 0. As conexões (pinos) de 0 a 13 são chamadas de portas digitais. Isso porque elas só funcionam emitindo ou recebendo um sinal fixo de 5 volts. Ou seja, é digital, pois na linguagem computacional utilizada, 0 volts representa o número 0 e 5 volts, o número 1. As conexões de *A0* a *A5* são portas analógicas. Estas são portas apenas para entrada e são do tipo analógica. Assim, recebem sinais de 0 a 5 volts. Ou seja, são infinitos os valores possíveis entre 0 e 5.

Então, como funciona este computador que não tem teclado nem monitor? Vejamos um exemplo, como o montado na Figura 1.9. Na montagem, temos uma caixa de som com um terminal ligado no *GND* (do inglês *ground*, ou seja, “terra” que é o **zero** volts) e o outro terminal ligado na porta 10 do Arduino. Além disso, temos um LED amarelo, com o terminal positivo ligado em um resistor (de cor azul)⁶ que está ligado na porta 4 do Arduino. Já o terminal negativo do LED está ligado diretamente no outro *GND*.

⁶ O resistor serve para fazer com que o LED não queime, pois os 5V da porta digital é muita voltagem para ele. No máximo, ele suporta uns 3 Volts.

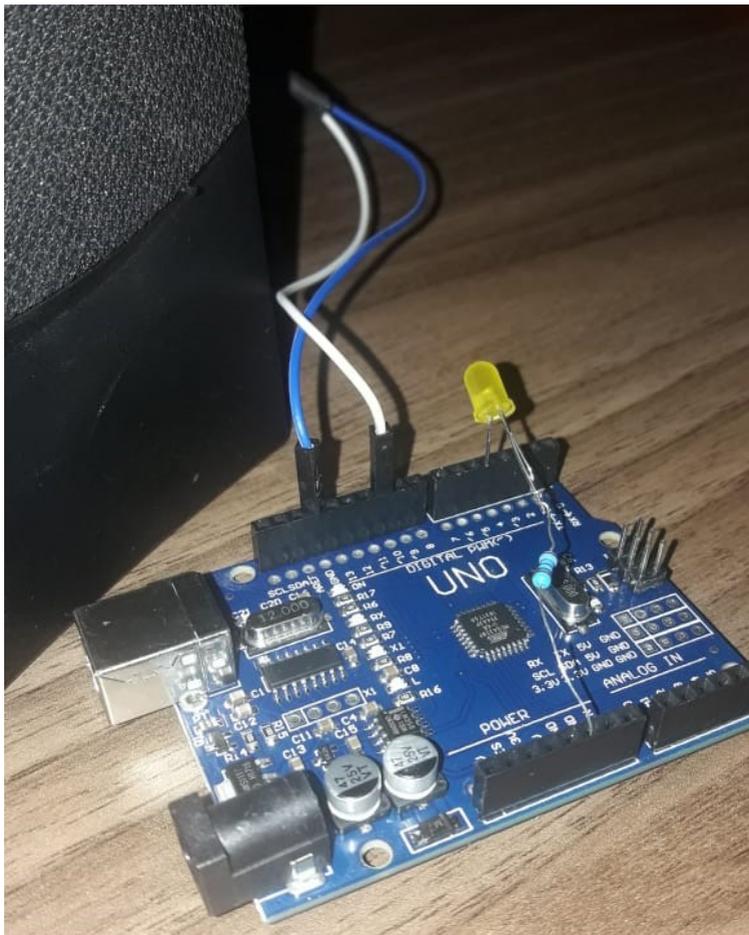


Figura 1.9 – Montagem de Arduino com caixa de som, resistor e LED.
Fonte: Dos autores.

Nesta montagem, podemos criar um programa que faça o Arduino mandar ligar a porta 4, fazendo com que o LED acenda. Além disso, poderíamos mandar o Arduino enviar um sinal sonoro pela porta 10 emitindo uma nota musical. O que fazer com isso depende da criatividade do usuário.

Noções super elementares sobre eletrônica e Arduino

Para evitar dissabores por parte de neófitos em eletrônica, com a queima de equipamentos durante a confecção do Celta, iremos indicar alguns rudimentos super elementares sobre eletricidade e eletrônica. **Leia com atenção!**

Os elementos da natureza são formados por átomos. Alguns possuem elétrons bastante livres que podem sair de um átomo e ir para outros átomos. Estes elementos são chamados de condutores elétricos. Em outros elementos, a passagem de elétrons é mais difícil, eles são chamados de isolantes elétricos. São exemplos de condutores os metais em geral e algumas soluções salinas. São exemplos de isolantes: borracha, cerâmica, plástico, vidro e o ar⁷.

Para que ocorra um **sinal elétrico** em um componente, são necessários **dois fios** a serem ligados a uma fonte de energia elétrica. Um, de onde vêm os elétrons e outro, para onde vão. Você pode notar isso nos equipamentos em geral, que possuem tomadas com dois pinos.

A questão é que a corrente elétrica é um movimento de elétrons, sempre de onde há mais elétrons para onde há menos elétrons, já que na natureza as coisas buscam o equilíbrio. Por exemplo, se você estiver descalço e pegar em um dos fios desencapados de uma tomada de sua casa irá levar um choque⁸. Isso porque no fio que você pegou existia uma quantidade de elétrons diferente da quantidade de elétrons da terra. Enquanto você não pegava no fio desencapado nada ocorria, depois que você pegou, ocorreu um sinal elétrico. Trata-se de uma máquina de choque elétrico. **NÃO** recomendamos que faça esta experiência, pois seria dolorosa, senão, fatal.

⁷ Observe que na verdade não existe isolante perfeito pois dependendo das condições sempre alguma condução de eletricidade pode existir, por exemplo, o ar é “isolante” mas não o suficiente para evitar um raio.

⁸ A bem da verdade, na tomada de 127 volts padrão de uma residência apenas um dos dois fios dá choque. O outro é a terra. Assim, você tem 50% de chance de ser eletrocutado neste caso.

Existem duas possibilidades de configuração da eletricidade nos fios. Uma é chamada de **corrente alternada** (AC, do inglês *alternating current*), que é a que tem na parede de sua casa. Na corrente alternada, cada pino da tomada fica variando a quantidade de elétrons 60 vezes por segundo, como é o padrão aqui no Brasil. A outra possibilidade de corrente é chamada de **corrente contínua** (DC, do inglês *direct current*) que é a que tem em uma pilha ou bateria, na saída de seu carregador de celular ou no Arduino.

Em **eletrônica, normalmente se usa corrente contínua** (DC). Ou seja, existe um terminal negativo (-), com mais elétrons e um positivo (+), com menos elétrons. Assim, uma pilha tem terminais + e -, uma saída de carregador de celular tem + e -, uma bateria de carro tem + e -, um cabo USB de computador tem + e -. Assim que algum componente elétrico for ligado entre estes terminais, iniciará uma corrente elétrica. Só lembre de nunca ligar o positivo no negativo diretamente, pois isso gerará um **curto circuito**, que pode levar a queima do equipamento, ou mesmo, a incêndios.

O carregador do celular transforma a corrente alternada (da tomada) em contínua, pois o celular utiliza componentes eletrônicos. Veja a Figura 1.10 com um carregador de celular indicando que na entrada é AC, ou seja, corrente alternada, de 100 a 240 volts, e de 50 a 60 vezes por segundo. Veja ainda que na saída ele indica DC, 5 volts.



Figura 1.10 – Carregador de celular. Fonte: Dos autores.

Em relação ao risco de choque elétrico, a DC é mais segura, pois nela a voltagem tende a ser bem menor. Ou seja, se você pegar em um fio na saída DC de seu carregador de celular, mesmo descalço, não sentirá o choque que levará, pois 5 volts não é percebido pelo seu corpo, em geral. Só não coloque na língua pois aí, sim, perceberá o choque.

Nos circuitos de corrente contínua (DC), o **negativo é chamado de GND** (do inglês *ground*), ou terra. A ideia é que no positivo existe uma menor quantidade de elétrons (que possuem carga negativa). Assim, para compensar, os elétrons vão do negativo para o positivo. Esta diferença resulta no que chamamos de voltagem (medida em volts, V).

Embora os elétrons caminhem do negativo para o positivo, **na prática dizemos que o sinal elétrico vai do positivo para o negativo**. Ou seja, do + para o -. O nome disso é sentido de corrente *convencional*. É mais prático pensar assim!

Alguns componentes eletrônicos possuem uma polaridade. Ou seja, devem ser ligados corretamente para não queimarem (positivo no positivo e negativo no GND), este é o caso

do LED. Outros, não possuem polaridade, como um resistor ou um autofalante. Então, tanto faz o terminal ligado.

O Arduino utiliza 5V para poder funcionar. A porta **USB** do computador fornece 5V. Um carregador de celular também fornece 5V. Porém, existem outras fontes que possuem voltagem diferente de 5V e você deve estar sempre atento a isso para evitar que o Arduino queime.

Existem várias formas de se alimentar o Arduino. Uma delas é utilizando o próprio **cabo USB** que fornece os 5V utilizados pelo Arduino. Esta já é a voltagem operacional do Arduino. Outra forma é com uma fonte de alimentação conectada na **entrada de alimentação do Arduino**. Nesse caso, pode-se ligar uma fonte DC entre 5V e 12V pois nesta entrada existe um circuito que faz a conversão para 5V.

No *Arduino Uno*, temos 3 pinos com a indicação **GND**. Ou seja, é o terra, o negativo, o -. Os 3 pinos são exatamente iguais, ou seja, tanto faz em qual você ligar, é a mesma coisa.

No *Arduino Uno*, temos um **pino** chamado **5V**. Ele é o sinal positivo de 5V. Muito **cuidado** onde for ligá-lo. Por exemplo, se você conectar um cabo entre o pino 5V e o GND você irá **QUEIMAR** o Arduino. Então, terá que jogar fora o Arduino e arcar com o prejuízo.

Os **pinos de 0 a 13** são pinos de entrada ou de saída. Ou seja, sua função varia conforme o programa. Da mesma forma que no pino 5V, nunca ligue diretamente um dos pinos de 0 a 13 no GND pois poderá queimar o Arduino, já que estes pinos também podem emitir 5V. A diferença é que o pino 5V emite o tempo todo, enquanto os outros apenas quando o programa mandar.

Para a **alimentação do Arduino**, temos várias possibilidades. Na Figura 1.11, vemos algumas delas: um cabo USB, uma bateria de 9V e uma fonte de alimentação de 6 a 12V DC.



Figura 1.11 – Alimentação para o Arduino. Fonte: Dos autores.

No jargão da tecnologia, utiliza-se os termos **sensores** e **atuadores**. Os sensores recebem dados ou ações do meio físico e os atuadores emitem dados ou ações ao meio físico. Assim, um LED e um autofalante são exemplos de atuadores, enquanto um microfone é um exemplo de sensor. A Figura 1.12 apresenta alguns botões, ou seja, são sensores, pois transformam a ação mecânica em uma sensibilidade elétrica. Em nosso projeto do Celta teremos como sensores os dois botões (*push buttons*) e como atuadores os seis motores que vão acionar a célula braille.



Figura 1.12 – Botões (*push-buttons*) diversos. Fonte: Dos autores.

Para o aprofundamento sobre o Arduino, indicamos os seguintes livros:

BLUM, Jeremy. *Explorando o Arduino: Técnicas e ferramentas para mágicas de engenharia*. Rio de Janeiro: Alta books. 2016.

MONK, Simon. *30 Projetos com Arduino*. Porto Alegre: Bookman, 2014.

1.4 O display braille⁹

Agora que entendemos um pouco mais a respeito do contexto sobre acessibilidade e sobre cultura maker, vamos voltar um pouco e rever a origem específica do braille, que foi nosso grande objeto de interesse neste projeto. Iniciemos com um breve histórico.

Em 1784, Valentin Hayu (1745-1822), considerado o “pai da educação dos cegos”, fundou em Paris a primeira escola de cegos (*Instituiton Nationale des Jeunes Aveugles*). Foi o criador de uma escrita em relevo, para permitir a leitura aos cegos, pelo tato, bem como de um *Ensaio sobre a educação dos cegos*. [...]

Louis Braille (1809-1852), um dos discípulos de Valentin Hauvy, entrou para o instituto de cegos de Paris com sete

⁹ Obs.: Partes do texto deste item já figuram em outras publicações dos autores relacionadas com este tema. A saber, o próprio projeto do edital enviado à CAPES e SANTIAGO, G. L. A.; BENGTON, C.; PINO, D.; PENDENZA, C.; KAWAKAMI, C. **Célula tátil Braille**: Desenvolvimento de uma Ferramenta da Acessibilidade. In: Anais do ESUD/CIESUD, 2019.

anos, quatro anos após ter ficado cego, foi músico distinto e depois professor no mesmo instituto. Entregou-se ao estudo de uma escrita para uso dos cegos, hoje universalmente conhecida por sistema Braille, cujas peculiaridades são descritas em seu Procédé pour écrire á l'usage des aveugles (Processo de escrever para uso dos cegos). [...]

Braille [...] criou, com seis pontos salientes, 63 combinações, que representavam letras simples e acentuações, pontuações, algarismos, sinais algébricos, contrações estenográficas e sinais para música. [...]

Com o impulso da escrita Braille, surgiram numerosas escolas para cegos (MIRADOR, 1987, p. 2207-2208).

Pensando na importância da inclusão de pessoas com deficiência visual, a célula tátil que foi desenvolvida se baseia especificamente no sistema braille, que é uma forma de codificação textual projetada para ser utilizada por pessoas com deficiência visual acentuada. A lógica que viabiliza o braille é o agrupamento de pontos na chamada célula braille. No Brasil, utiliza-se o sistema braille de 6 pontos, que consiste em duas colunas de três pontos, possibilitando 63 sinais distintos. Vide a Figura 1.13.

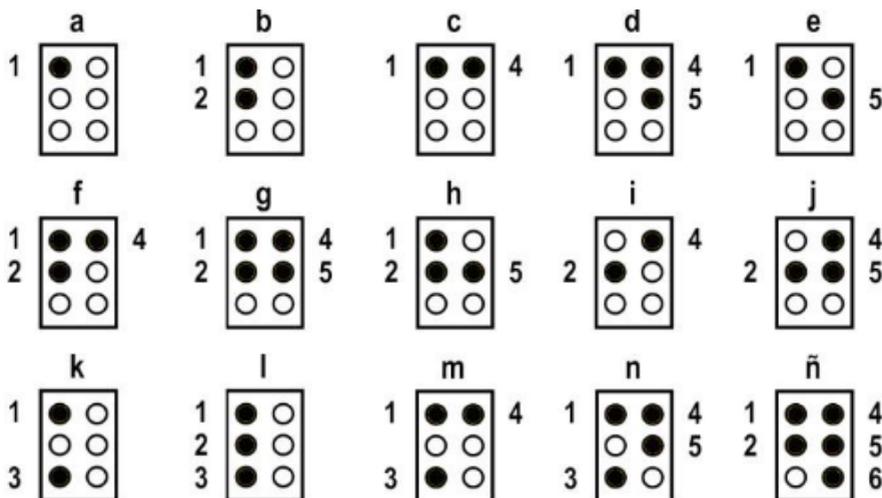


Figura 1.13 – Braille de 6 pontos.
Fonte: Instructor braille (2018).

O sistema braille pressupõe que o indivíduo utilize o tato e, com os dedos, perpassa da esquerda para a direita diversas células, cada uma contendo uma das 63 possibilidades. Dessa forma, é necessário um meio físico, tradicionalmente papel com pontos em alto relevo. Para realizar esses padrões, utiliza-se uma reglete¹⁰, uma máquina de escrever braille, uma impressora braille ou até um display braille. De todas essas possibilidades, os dois últimos são os mais recomendados para a elaboração de conteúdos em massa, como os indicados para a EaD, porém, são equipamentos de elevado valor comercial, o que dificulta sua aquisição pela pessoa com deficiência ou pelas instituições para uso em polos de apoio presencial na modalidade a distância. Para a EaD, o display braille é mais recomendado, pois viabiliza a leitura imediata do conteúdo digital, que é preponderante em cursos dessa modalidade.

O display braille é um hardware que pode ser definido como um periférico de computador de saída tátil que permite a auto modelagem de padrões de pontos utilizados para viabilizar a leitura no sistema Braille. Vide a Figura 1.14, que apresenta um display braille com 80 células no sistema americano de 8 pontos.



Figura 1.14 – Display braille. Fonte: Dos autores.

O display braille só deixa de ser uma grande possibilidade para a inclusão devido ao seu elevado custo, chegando a custar vinte vezes mais do que um computador convencional. Com exceção do preço, o display braille é uma alternativa muito

¹⁰ Trata-se de uma espécie de régua para guiar a pontuação em um papel para a escrita braille.

adequada e até mais ecológica, considerando a economia de papel e energia.

Por isso, nossa proposta visou desenvolver uma alternativa de difusão de um tipo de display braille, customizado e de baixo custo. A seguir, descreve-se esta proposta de projeto que foi executado.

1.5 A proposta do projeto Sistema de célula tátil para leitura braille¹¹

1.5.1 Resumo da proposta

Nesta proposta, visamos contribuir para o desenvolvimento tecnológico educacional brasileiro de forma inventiva e inovadora ao elaborar o projeto e a prototipagem de um sistema (software e hardware), aberto e de baixo custo, para leitura de textos em formato digital, que consiste de célula tátil para viabilizar a leitura por pessoas com deficiência visual ou com surdocegueira, prioritariamente em braille, porém com a possibilidade de envolver letras em alto relevo. Além disso, este projeto tem que prever aberturas para aprimoramentos e desdobramentos futuros incluindo teclado braille, interface com leitor de tela; bluetooth, gerar conhecimentos para a criação de outros objetos acessíveis utilizando-se os mesmos princípios desta missão (baixo custo, fácil implementação e replicação). O hardware resultante trata-se de uma caixa com conexão USB para o computador, que permite a leitura em braille de um texto do computador. O usuário mantém um dos dedos no dispositivo para realizar a leitura tátil em braille.

¹¹ Obs.: Partes do texto deste item já figuram em outras publicações dos autores relacionadas com este tema. A saber, o próprio projeto do edital enviado à CAPES e SANTIAGO, G. L. A.; BENGTSON, C.; PINO, D.; PENDENZA, C.; KAWAKAMI, C. **Célula tátil Braille**: Desenvolvimento de uma Ferramenta da Acessibilidade. In: Anais do ESUD/CIESUD, 2019.

1.5.2 Público-alvo da proposta

O público-alvo do projeto foi: 1. Pessoas com deficiência visual; 2. Instituições de ensino; 3. Fundações; e 4. Empresas e Organizações.

1.5.3 Problemas e soluções considerados na proposta

A seguir, elencamos alguns problemas e dilemas que nos levaram a pensar neste projeto:

1. Dadas as circunstâncias, muitos estudantes com deficiência visual, em vez de realizar a leitura de textos no computador através dos leitores de tela, poderiam realizar a leitura em braille.
2. Pessoas com surdocegueira não podem utilizar o leitor de tela e dependem preponderantemente da leitura tátil.
3. Pessoas cegas com mobilidade reduzida podem não conseguir uma boa fluidez na leitura na linha braille convencional, já que esta demanda movimentos finos dos dedos e das mãos. Isso pode ser um obstáculo para a leitura do estudante.
4. O alto custo da linha braille convencional reduz seu uso popular, ou seja, gera exclusão, já que muitos não terão acesso a esse recurso.
5. Hardwares e softwares em geral são de proprietários, dificultando e encarecendo o uso popular.
6. Alternativas ao uso da linha braille são escassas e pouco difundidas.
7. Células braille eletrônicas só são vendidas no mercado internacional e são de difícil aquisição e de difícil montagem em um dispositivo de leitura prático.
8. Células e linhas braille eletrônicas convencionais muitas vezes possuem 8 pinos, diferente do padrão Braille utilizado no Brasil que são 6 pinos.
9. Como indicado no pressuposto do edital nº 3/2018 - Ferramentas de acessibilidade, as instituições de ensino devem fazer de tudo para garantir a acessibilidade em suas

propostas e infraestrutura educacional, porém ainda resta muito a ser realizado e a cada dia verifica-se uma maior demanda por recursos acessíveis.

Diante desses apontamentos, listamos os principais pontos acerca da relevância deste projeto:

1. Nos últimos anos, ocorreu um aumento no processo de inclusão das pessoas com deficiência na educação e mais recentemente na Educação a Distância, como nos cursos da UAB. Dessa forma, ocorreu a necessidade de produzir materiais didáticos acessíveis variados e em maior escala. Este projeto visa integrar-se a esse processo de aprendizagem dos estudantes com deficiência visual, surdocegueira e mobilidade reduzida. A produção de um protótipo de baixo custo e nacional da linha braille permite o acesso ao conhecimento.
2. As linhas braille comercializadas no país não possuem origem local, ou seja, são importadas, tendo um elevado valor de comercialização comparada a renda de boa parte da população com deficiência no Brasil. Portanto, para que mais pessoas possam ter acesso a este equipamento foi pensado um produto similar que tenha um menor custo e possa ser acessível às pessoas com deficiência.
3. Pretendemos deixar as informações de replicação abertas com o objetivo de que outras pessoas possam reproduzi-lo, já que disponibilizaremos gratuitamente o modo de fazer, todos os materiais e informações para organismos diversos. Isso ajudará outras instituições de ensino a oferecerem acessibilidade para os seus estudantes.

1.6 Histórico do desenvolvimento do projeto¹²

O projeto foi concebido, desenvolvido e finalizado entre os anos de 2018 a 2020. A Figura 1.15 apresenta as diversas etapas

¹² Obs.: Partes do texto deste item já figuram em outras publicações dos autores relacionadas com este tema. A saber: SANTIAGO, G. L. A.; BENGTON, C.; PINO, D.; PENDENZA, C; KAWAKAMI, C. **Célula tátil Braille: Desenvolvimento de uma Ferramenta da Acessibilidade**. In: Anais do ESUD/CIESUD, 2019.

que serão consideradas no desenvolvimento do projeto. Observa-se que na fase 1 nos dedicamos ao refinamento dos objetivos, ou melhor, das especificações do sistema. Observa-se, ainda, pela linha tracejada vermelha, que esta fase ocorreu em um movimento cíclico em relação à fase posterior, que é a montagem das equipes e aquisição dos insumos. Isso se deve ao fato de que com novos elementos no grupo e novos equipamentos, ideias e possibilidades adicionais surgiram. Após estas fases, tivemos as fases 3 e 4, nas quais ocorreram diversos ciclos iterativos de prototipagem e testes de conceitos. Ao final da fase 4, após alguns ciclos, ocorreu a validação de cada protótipo que foi escolhido como produto final que foi divulgado (na fase 5).

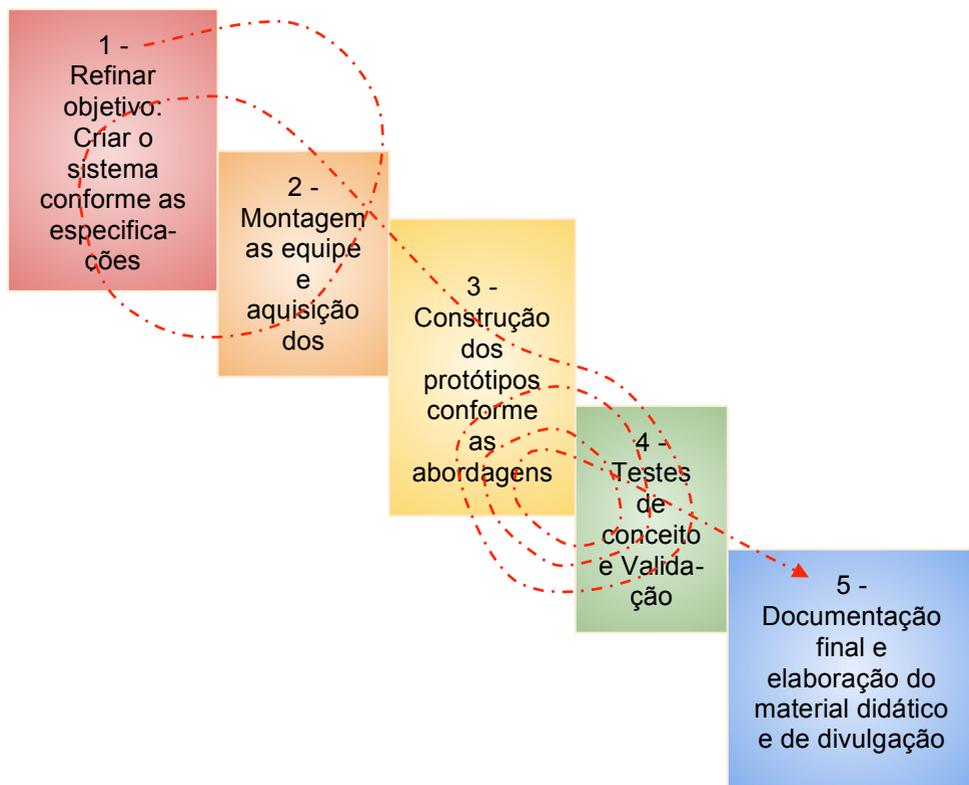


Figura 1.15 – Fases do projeto. Fonte: Dos autores.

1.6.1 Formação da equipe

A equipe constituiu-se de forma multidisciplinar, dada a natureza do projeto. Foi formada pelo Coordenador do Projeto, uma Gerente de Projetos, dois profissionais da área de Engenharia da Computação (um para o software e outro para o desenvolvimento do hardware), uma profissional da linguagem responsável pelo aporte ao trato com o braille, três educadoras que auxiliaram o projeto nas questões que envolvem aprendizagem e uma administradora pública que atende às tarefas administrativas e financeiras perante a Fundação Institucional da UFSCar e a Capes. A Figura 1.16 ilustra este organograma.



Figura 1.16 – Organograma da equipe de trabalho do projeto.
Fonte: Dos autores.

A bem da verdade, ocorreram mudanças entre a equipe que foi responsável pela elaboração inicial do projeto e a equipe que de fato contribui na execução. Nesta documentação, incluímos apenas a equipe que atuou na execução do projeto, conforme indicado no quadro 1.3.

Quadro 1.3 – Membros da equipe de execução

Coordenação	Glauber Santiago
Gestão	Clarissa Bengtson
Hardware	Cristian Pendenza
Software	Juan Santos
Linguagem	Douglas Pino
Educação	Nadia Perez Pino Rafaela Marchetti Meire Moreira Cordeiro
Administração	Vanessa Onody

1.6.2 Primeiros protótipos

Na concepção do projeto, não tínhamos muita noção das dificuldades e limitações técnicas que seriam enfrentadas. Por isso, a curva de aprendizagem foi bastante acentuada neste processo. A ideia inicial baseava-se em 4 abordagens, como seguem:

Na **abordagem 1**, a tecnologia seria baseada na utilização de uma grande régua na qual cada caractere seria impresso previamente. Esta abordagem aparentava ser de fácil implementação e propiciaria a possibilidade da utilização de alto relevo. A Figura 1.17 indica o princípio de funcionamento: o usuário manteria o dedo fixo no orifício quadrangular, acima da base em vermelho. Então, a régua verde se posicionaria no caractere a ser lido e seria deslocada para cima, permitindo a leitura tátil. Em seguida, a régua seria deslocada novamente para baixo e o próximo caractere seria posicionado, e assim por diante. Isso sempre seguindo o texto a ser lido que estaria no computador.

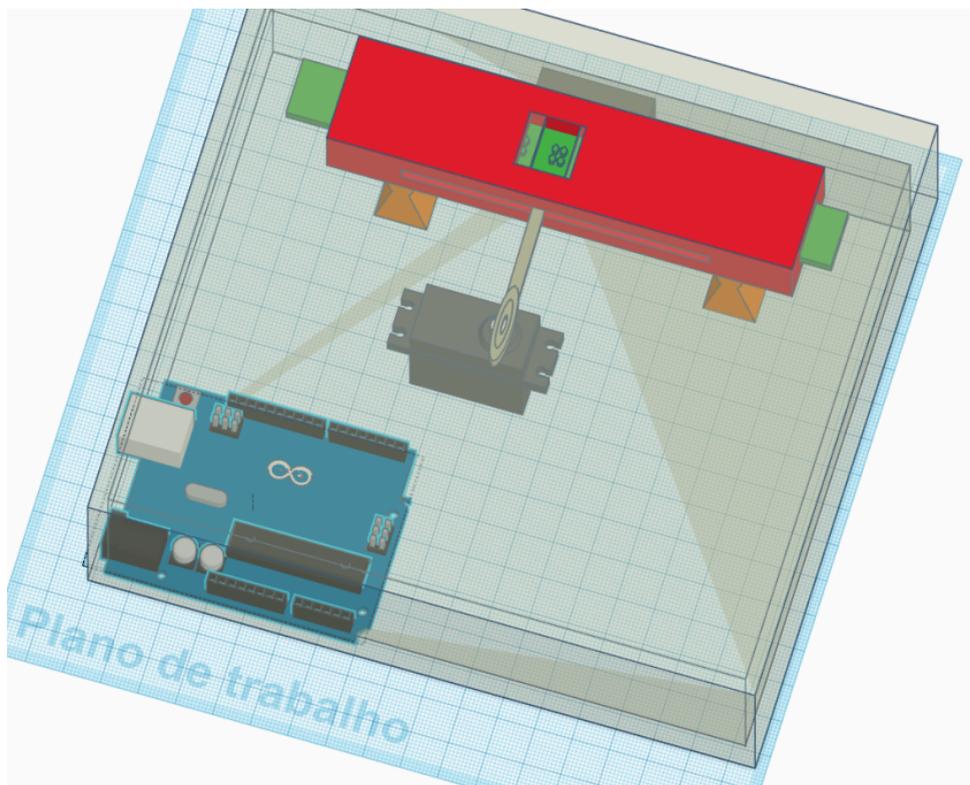


Figura 1.17 – Abordagem 1: Régua retangular com padrões pré-estampados. Fonte: Dos autores.

A **abordagem 2** era semelhante à anterior, porém, no lugar de uma régua seria um disco. Na Figura 1.18 pode-se visualizar o orifício quadrangular no qual o usuário iria manter seu dedo. Observa-se ainda na figura que o mesmo caractere poderia ser tateado na escrita braille e na escrita em alto relevo.

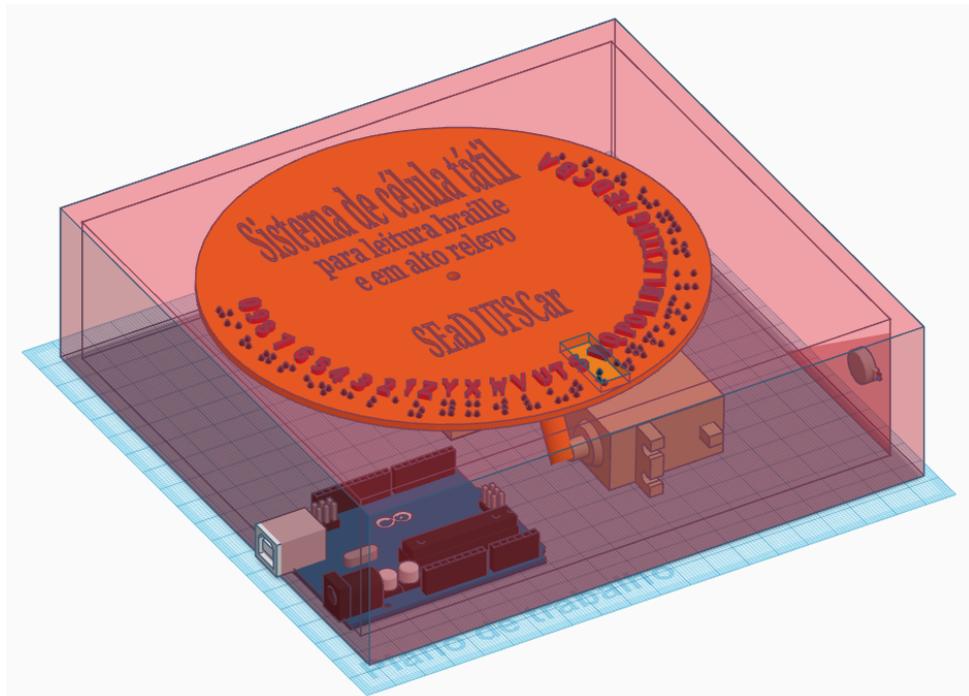


Figura 1.18 – Abordagem 2: Disco único com padrões pré-estampados.
Fonte: Dos autores.

A **abordagem 3** e a **abordagem 4** (Figuras 1.19 e 1.20) se caracterizam pela montagem dos pontos das células, diferentemente das anteriores, nas quais os caracteres seriam pré-estampados. Com esta montagem, se pretendia reduzir o tamanho do disco ou da régua, já que a combinação seria montada conforme o caractere a ser exibido. Os desafios dessas abordagens se referem ao modo de afastamento dos pontos do dedo do usuário entre cada mudança de caractere.

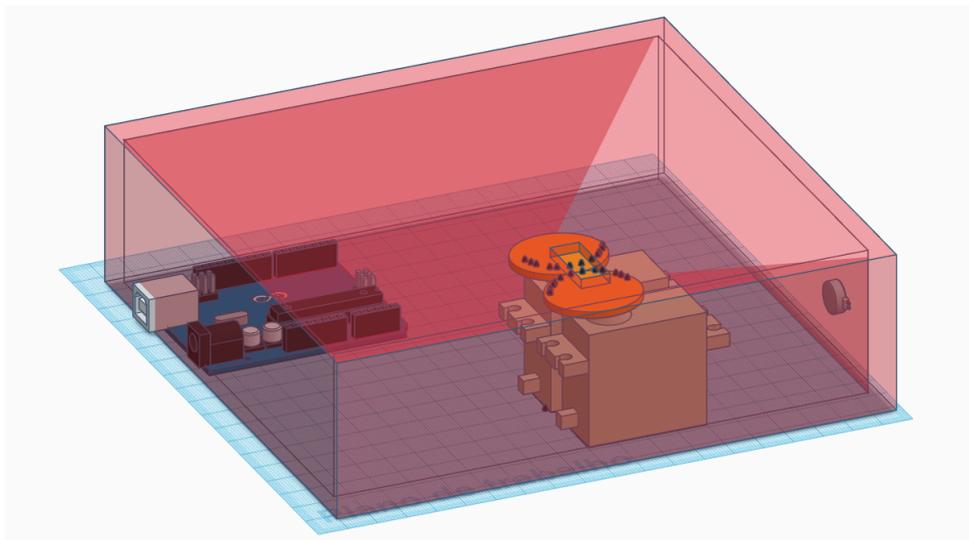


Figura 1.19 – Abordagem 3: Discos múltiplos para montagem de combinações de pontos. Fonte: Dos autores.

Observe na Figura 1.20 que temos dois discos, cada um com algumas combinações de pontos. Ambos, em conjunto, possibilitariam a geração de qualquer caractere braille. Na Figura 1.20, temos a visualização da abordagem 4, com régua. Cada uma com uma variedade de pontos que, em conjunto, formam um caractere braille.

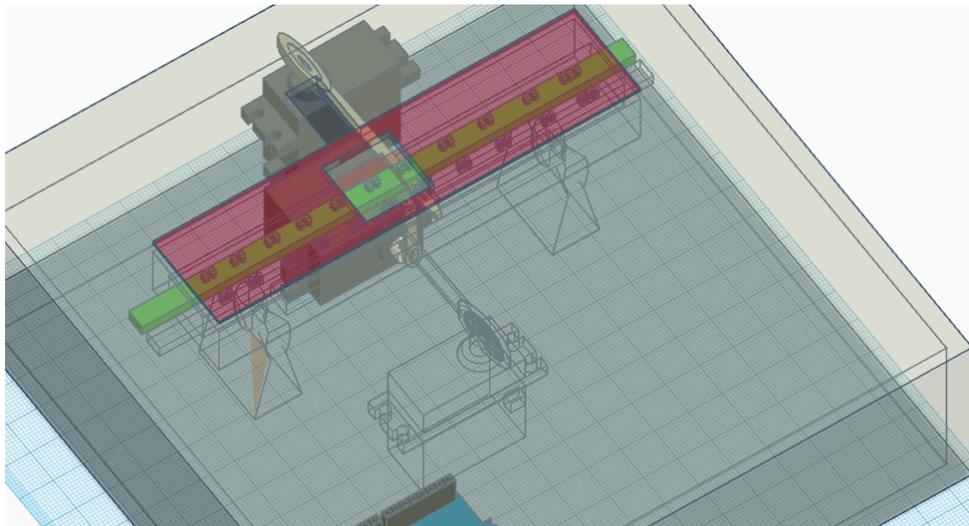


Figura 1.20 – Abordagem 4: Régua retangulares múltiplas para montagem de combinações de pontos. Fonte: Dos autores.

De todas essas abordagens, apenas a primeira foi executada completamente e o protótipo construído pode ser visualizado na Figura 1.21.

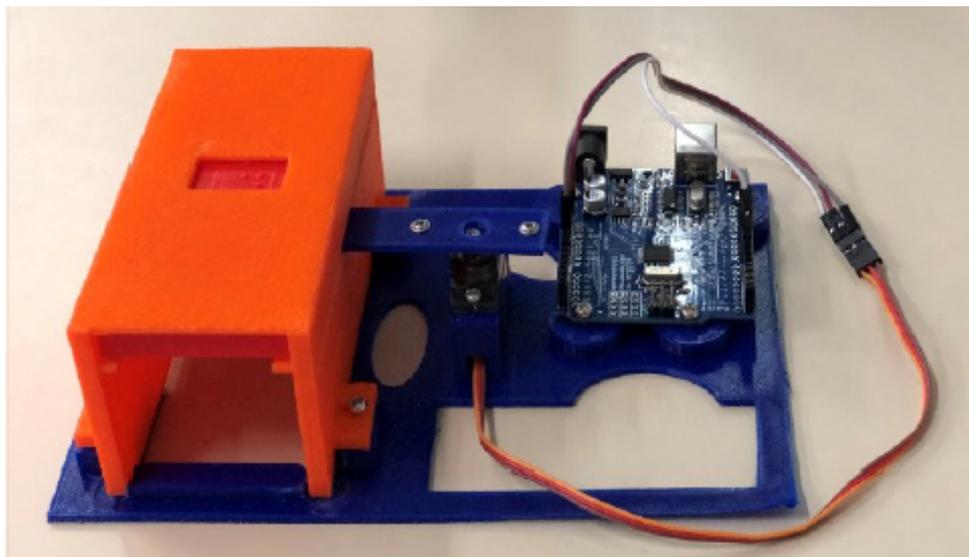


Figura 1.21 – Fotografia do protótipo na abordagem 1. Fonte: Dos autores.

Ao elaborarmos a abordagem 1, vimos que ela só seria factível para um número distinto de caracteres, 11, no caso. Sendo que para o braille completo necessitaríamos de cerca de 6 vezes esta quantidade. Porém, a abordagem pode ser utilizada para o manejo apenas de números em braille.

Para assistir um vídeo com a abordagem 1 sendo utilizada, acesse o seguinte link:

<<https://www.youtube.com/watch?v=UYjkagUZ6Yg>>.

A abordagem 2 foi iniciada, todavia verificamos que a circunferência do disco teria que ser muito grande, e isso geraria uma fragilidade no motor (servo motor) principal. Ou seja, com o uso teríamos um provável desgaste, resultando em ineficiência do produto.

As abordagens 3 e 4 não foram executadas, mas, inspirada nelas foi elaborada uma nova abordagem, a **abordagem 5**. As Figuras 1.22 e 1.23 apresentam algumas fotos com peças impressas para esta abordagem.

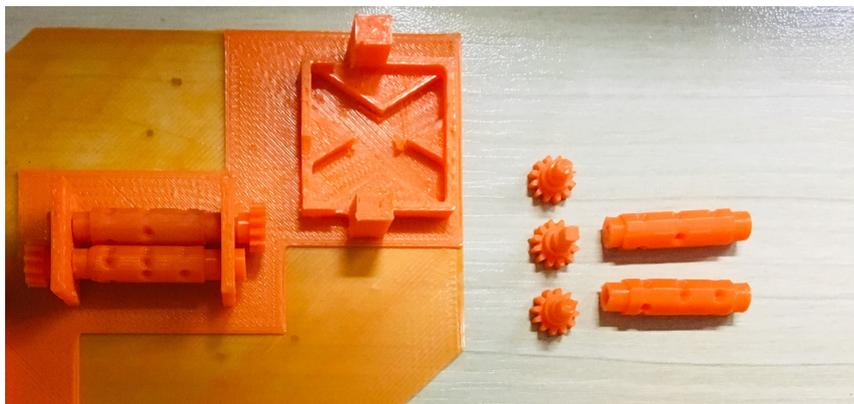


Figura 1.22 – Algumas peças para a abordagem 5. Fonte: Dos autores.



Figura 1.23 – Servo e motor com engrenagem para a abordagem 5.
Fonte: Dos autores.

A abordagem 5 também não logrou êxito pois as peças tinham que ser muito pequenas e rígidas para um funcionamento eficaz. Elas teriam que ser metálicas e não poderiam utilizar tecnologias fáceis e baratas de usar, como era a proposta do projeto. Então, buscou-se algo da ideia das abordagens 3 e 4 e principalmente do princípio dos displays braille convencionais. Esta foi nossa **abordagem 6**, que foi a que obteve sucesso e aprovação pela equipe e pelos testadores.

A ideia da abordagem 6 é que por meio de pequenas alavancas cada um dos pontos da célula é movimentado independentemente. O desafio é que se utilizam 6 micromotores, independentes, para este acionamento. No final das contas, esta miniaturização possibilitou um resultado melhor do que o esperado, com a obtenção de uma célula bem semelhante aos modelos comerciais.

Tudo o que será apresentado a seguir refere-se a esta sexta abordagem, que foi a que escolhemos para viabilizar o projeto.

1.6.3 Realização do protótipo final

Para a produção do hardware, utilizou-se da tecnologia do Arduino, que é um hardware livre em relação a direitos de uso. Além disso, valemos de processos de impressão em impressora 3D para a construção das peças plásticas. O hardware conta com peças e fiações simples, que podem ser facilmente adquiridas por

interessados em replicar a proposta. A Figura 1.24 apresenta o aspecto do protótipo, quando estava em testes.



Figura 1.24 – Protótipo desenvolvido de uma célula braille de 6 pontos, em tamanho padrão e já funcional. Fonte: Dos autores.

A fase de elaboração do hardware/software considerou os seguintes elementos:

1. Célula tátil impressa em 3D, acionada por seis micromotores DC 5V;
2. Software desenvolvido em C++ para controlar o Arduino; e
3. Módulo eletrônico para controle e inversão dos motores de acionamento da célula, utilizando 3 CIs (circuitos integrados), do tipo L293D (circuito ponte-h) e 2 botões de apertar (*push bottons*).

A Figura 1.25 apresenta a célula tátil e o módulo de controle montados ainda na protoboard. Ao fundo, à esquerda, a placa azul é o Arduino uno R3 utilizado. Ao centro, uma protoboard, branca com diversos cabeamentos e alguns componentes eletrônicos. À esquerda, outra protoboard com cabeamentos, e ao lado desta a célula braille construída e que utiliza seis micromotores, como os utilizados na vibração em smartfones. Entre os componentes eletrônicos, temos 3 circuitos integrados (CI) L293D que possuem duas portas cada um deles para fazer a inversão dos motores.

Assim, quando um motor gira em um sentido do pino ao qual está conectado na célula sobe e quando gira para o outro lado ele desce.

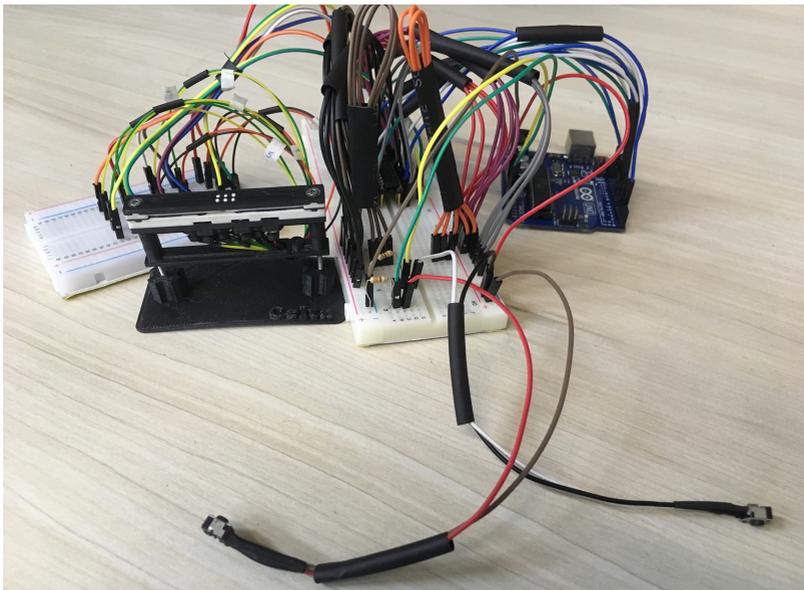


Figura 1.25 – Protótipo funcional do Celta na protoboard.
Fonte: Dos autores.

Na parte inferior da mesma figura, veem-se dois botões. Esses são os botões que avançam caractere e voltam, ou que avançam palavra. Vamos supor que se tenha um texto na área de trabalho do computador. Então, copia-se este texto (Ctrl+c) e, para colar o texto no software do Celta, basta segurar os dois botões. Depois que o texto está lá basta iniciar a execução do software e usar os botões para frente e para traz para ir mudando o caractere conforme está na tela. Ao segurar por 2 segundos, o software pula para a próxima palavra.

Para mais informações, assista ao vídeo *Protótipo em uso* acessando o seguinte link:

<<https://www.youtube.com/watch?v=snuEJvb7nf4>>.

2. Utilização do Celta

Como indicamos no capítulo anterior, o Celta foi concebido para ser uma alternativa de acessibilidade (financeiramente falando) aos displays braille comerciais. Porém, contendo atualmente apenas uma Célula é mais indicado para atividades de aprendizado inicial do braille e menos para a leitura extensiva. Neste capítulo, iremos apresentar algumas ideias de utilização do Celta.

Na Figura 2.1 vemos uma das possibilidades de se criar materiais em braille para atividades didáticas, porém a impressora braille é muito cara em termos financeiros e ainda muito ruidosa. Além de gastar papel e ainda ser lenta para imprimir.

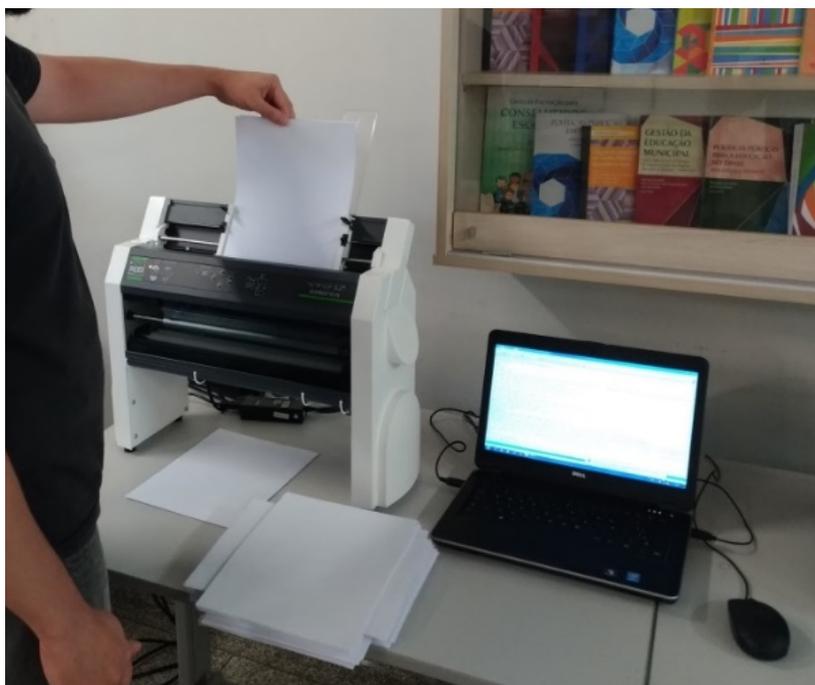


Figura 2.1 – Impressão em braille. Fonte: Dos autores.

De qualquer forma, a impressora braille é muito interessante pois permite o armazenamento de material produzido e a facilidade

e utilização pelo usuário. Conforme se observa na Figura 2.2 com um usuário lendo em braille.

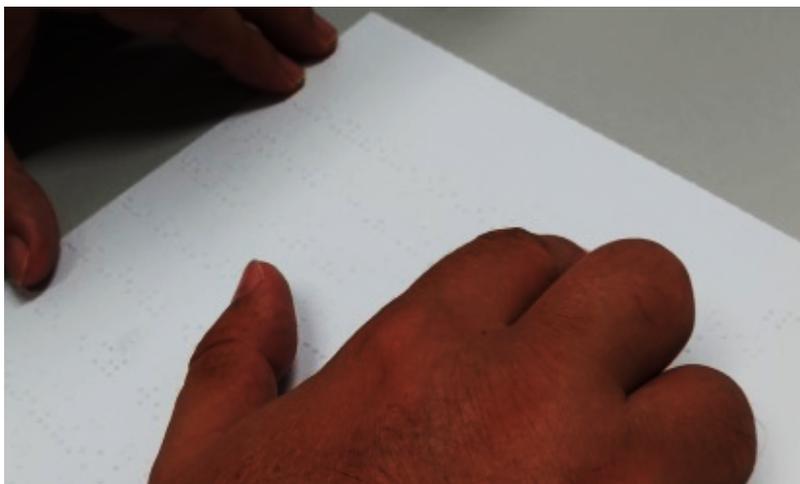


Figura 2.2 – Material em braille. Fonte: Dos autores.

Para a criação de material em braille, uma alternativa de baixo custo é a reglete (ver Figura 2.3). Com ela, a escrita braille pode ser facilmente realizada de maneira que o professor pode preparar material de forma bastante próxima das demandas do estudante.

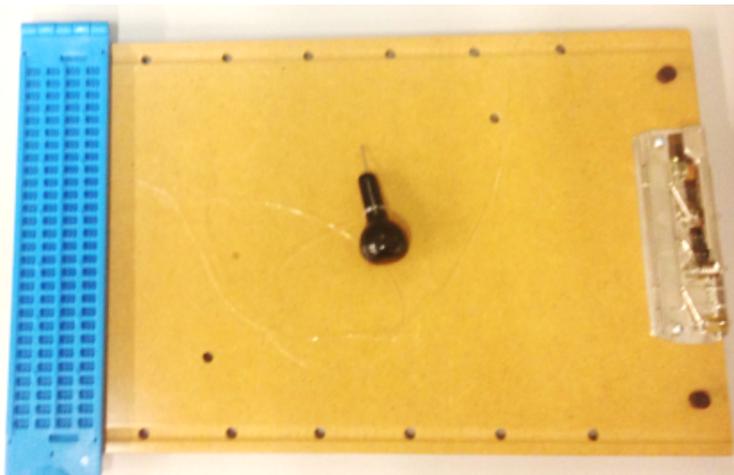


Figura 2.3 – Reglete. Fonte: Dos autores.

Então, uma outra alternativa para a preparação de materiais didáticos seria o display braille ligado ao computador, que é o objeto do nosso projeto. Assim, como dissemos, o Celta, em sua versão atual, possui apenas uma célula (vide Figura 2.4), enquanto outros displays braille possuem de 10 a 80 células. Sendo assim, sua usabilidade intensiva para leituras longas é menos indicada pois o usuário tenderá a ter uma menor velocidade de leitura. Porém, para leituras breves e para o aprendizado, o Celta de uma célula é muito bem aplicável.



Figura 2.4 – Fotografia do Celta. Fonte: Dos autores.

A utilização de uma única célula do aprendizado inicial em braille não é nenhuma novidade, então o Celta pode ser utilizado nesta etapa para exercitar a sensibilidade tátil e a memorização:

- da exploração espacial geral dos pontos;
- da primeira série (letras de “a” a “j”);
- da segunda série (letras de “k” a “t”);
- da terceira série (letras de “u”, “v”, “x”, “y”, “z”, “ç”, “é”, “á” e “ú”);
- da quarta série (“â”, “ê”, “ô”, “@”, “à”, “í”, “ü”, “õ” e “w”);
- da quinta série (““, “,“, “:“, “.“, “?”, “!“, “(“, “)“, aspas, “_”);
- da sexta série (“í”, “ã”, “ó”, sinal de número, apóstrofo, e “-“; e

- da sétima série (grifo, letras maiúsculas e \$).

2.1 Exercícios com o Celta

Com o objetivo de facilitar a introdução do uso do Celta, elaboramos uma série de exercícios que pode ser utilizada no aprendizado inicial do braille. Veja os itens a seguir!

2.1.1 Exercícios iniciais com uso do Celta

Seguem algumas sugestões de textos para treinamento de leitura no Celta em cada uma das séries de treinamento¹³.

Para a exploração inicial da célula, o site <http://www.deficienciavisual.pt/txt-iniciacao.htm> apresenta uma sequência de passos muito interessante, que adaptamos e transformamos em exercícios. Vejamos cada um deles no Quadro 2.1:

Quadro 2.1 – Sugestões de exploração inicial da célula (Fonte: adaptado de Ekvall et al. 2020).

Descrição	pontos	Caractere para o exercício do Celta
Célula.	nenhum	espaço
Fila da esquerda em acima.	1	a
Fila da esquerda ao meio.	2	,
Fila da esquerda em baixo.	3	'
Fila da esquerda completa.	123	l
A fila da esquerda não está completa. Falta o ponto de baixo.	12	b
A fila da direita não está completa. Falta o ponto de baixo.	45)
A fila da esquerda não está completa. Falta o ponto de cima.	23	;

¹³ Arquivos com estes exercícios está disponíveis no site do projeto, no link: <https://celta.ufscar.br/repositorio-de-arquivos/>.

A fila da esquerda não está completa. Falta o ponto do meio.	13	k
A fila da direita não está completa. Falta o ponto do meio.	46	\$
Um ao lado um do outro, em cima.	14	c
Um ao lado um do outro, no meio.	25	:
Um ao lado um do outro, em baixo.	36	-
Dois juntos, mas desalinhados.	15	e
Dois juntos, mas desalinhados.	26	?
Dois juntos, mas desalinhados.	24	i
Dois juntos, mas desalinhados.	35	*
Dois longe, mas desalinhados.	16	1
Dois longe, mas desalinhados.	24	ì

Exercício 1¹⁴ – Leia, no Celta, cada um dos caracteres e indique quais os pontos estão sendo utilizados e a descrição, conforme indicado no quadro anterior:

a, 'la, 'la, 'la, 'l

b); k\$ b); k\$ b); k\$

c:- c:- c:- c:-

e? l* e? l* e? l*

1 ì 1 ì 1 ì

Exercício 2 – Idem ao anterior:

'l, a b); k\$ a c:-, 'le? l* 1 ì

) b); k 1 ì a, 'la, 'l\$ b); k\$

e? l c* e?: - c:- c:- 1 ì l*

2.1.2 Primeira série (letras de "a" a "j")

Exercício 1 – Leia, no Celta, cada um dos caracteres (a ou b):

¹⁴ Para utilizar esses exercícios, basta copiar o conteúdo em azul para o software do Celta e iniciar a leitura braille.

a a b b ab ab

Exercício 2 – Leia, no Celta, cada um dos caracteres (a ou b):

ba ba

Exercício 3 – Leia, no Celta, cada um dos caracteres (a ou c):

c c ac ac

Exercício 4 – Leia, no Celta, cada um dos caracteres (a, b ou c):

abc abc abc abc abc abc abc abc abc abc

Exercício 5 – Leia, no Celta, cada um dos caracteres (agora tem mais o d):

d d abcd abcd abcd abcd abcd abcd abcd

Exercício 6 – Leia, no Celta, cada uma das palavras:

dada dada cada cada ada ada baca

Exercício 7 – Leia, no Celta, cada um dos caracteres (de a até e):

e e abcde abced abced abacadaeabacadae

Exercício 8 – Leia, no Celta, cada um dos caracteres (agora tem mais o f):

f f af af abcdef f f afbfcfdfef acbdef

Exercício 9 – Leia, no Celta, cada um dos caracteres (agora tem até o g):

g g ag ag abcdefg g g agbgcgdgegfg acbdefg

Exercício 10 – Leia, no Celta, a seguinte frase:

a faca de cada fada

Exercício 11 – Leia, no Celta, cada um dos caracteres (incluído o h):

h h abcdefgh fgh abc ah gh acegh bdfh ahbh

Exercício 12 – Leia, no Celta, cada um dos caracteres (agora tem o i também):

i j abcdefghi ai bi ci ihgfedcba di ei fi gi hi aiai

Exercício 13 – Leia, no Celta, cada um dos caracteres (de a até j):

j j abcdefghij aj bj cj jihgfedcba dj ej fj gj hj ajci

Exercício 14 – Leia, no Celta, a seguinte frase:

a jaca da dede

2.1.3 Segunda série (letras de “k” a “t”)

Exercício 1 – Leia, no Celta, cada um dos caracteres (k, l, m, n, e o):

k l m n o k l m n o k l m n o n m l k o n m l k m o k l

Exercício 2 – Leia, no Celta, cada um dos caracteres (p, q, r, s e t):

p q r s t p q r s t p q r s t s r q p t s r q q t r p

Exercício 3 – Leia, no Celta, cada um dos caracteres (de k a t):

k l m n o p q r s t t s r q p o n m l k s m n o p k t r

Exercício 4 – Leia as seguintes palavras (com as letras de a a t):

casa bala bola bolo elefante jaboti mala campo capim corre melado
oi janela pipoca jogo acelora equipe time todo mundo caber saber
google computador Internet braille amigos som onda pecado santo

Exercício 5 – Leias as seguintes frases:

a casa dos meus amigos foi feita de cimento e de tijolos
sempre busco ser bom para os meus companheiros de aula
estou usando o celta para ler o braille

2.1.4 Terceira série (letras de “u”, “v”, “x”, “y”, “z”, “ç”, “é”, “á” e “ú”)

Exercício 1 – Leia, no Celta, cada um dos caracteres (u, v, x, y, e z):

u v u v x y x y z z y x u v y y x x v x x u y z x y u z

Exercício 2 – Leia, no Celta, cada um dos caracteres (incluindo ç, é, á, e ú):

ç é á ú ç ç é é á á ú u ç é á ú ç ç é é á á ú u ç é á ú ç ç é é á á ú u

Exercício 3 – Leia, no Celta, as seguintes frases:

até que enfim eu aprendi a utilizar alguns acentos como o acento agudo e também já posso usar o cedilha que é ç

eu estou achando bom ler estas coisas até agora.

2.1.5 Quarta série (“â”, “ê”, “ô”, “@”, “à”, “í”, “ü”, “õ” e “w”)

Exercício 1 – Leia, no Celta, cada um dos caracteres a mais (“â”, “ê”, “ô”, “@”, “à”, “í”, “ü”, “õ” e “w”):

a à â e è ê o ò ô õ @ w a à â e è ê o ò ô õ @ w a à â e è ê o ò ô õ @ w

Exercício 2 – Leia as seguintes frases

hoje eu vou mandar um email e por isso aprendi o símbolo da arroba que é @

walita é uma marca de eletrodomésticos.

revoluções são formadas por revolucionários.

2.1.6 Quinta série (“,” , “;”, “:”, “.”, “?”, “!”, “(,)”, aspas, “ _ ”

Exercício 1 – Leia a seguinte frase:

A vírgula é , o ponto e vírgula é ; o ponto é . o ponto de interrogação é ? o abre aspas é “ o fecha aspas é igual “ o sublinhado é _ o ponto de exclamação é ! mas está difícil escrever exclamação sem utilizar o til que vou aprender logo, na sexta série.

2.1.7 Sexta série (“í”, “ã”, “ó”, sinal de número, apóstrofo, e “-”)

Exercício 1 – Leia os seguintes caracteres:

í ã ó ‘ - Í ã ó ‘ - Í ã ó ‘ - a à á â ã o ó ô î í – ‘

Exercício 2 – Leia a seguinte frase:

agora que eu sei ler o til na letra a dá para escrever direito a exclamação! e também já sei os números 1 2 3 4 5 6 7 8 9 e 0

2.1.8 Sétima série (grifo, letras maiúsculas e \$)

Exercício 1 – Leia, no Celta, os seguintes caracteres

aA bB cC dD eE fF gG hH il jJ

aA1 bB2 cC3 dD4 eE5 fF6 gG7 hH8 il9 jJ0

Maria Paula Joana Kelly R\$ 300,00

2.2 Utilizando o Celta virtual para treinamento de não usuários do braille

Um dos aspectos a serem considerados nos processos de ensino-aprendizagem do braille a é necessidade de pessoas que não são cegas terem também um aprendizado do braille. O objetivo é a criação de canais de comunicação entre usuários e não usuários do braille. Por exemplo, em uma escola o professor de matemática da sala de aula convencional que não utiliza o braille pode se comunicar melhor com os alunos cegos e com o professor da sala de recursos, se tiver uma maior familiaridade com o braille.

Considerando isso, no âmbito do projeto do Celta investimos um tempo na elaboração de uma versão *on-line* e visual do Celta chamada de *Prática Virtual Celta - Treinamento Braille*. A Figura 2.4 apresenta a tela principal do software. O software é basicamente um aplicativo de treinamento, no qual ele fala (verbalmente) uma letra ou caractere e o usuário deve escolher a configuração de pontos correta para o que foi pedido. Quando o usuário acerta, é dito que ele acertou e parte-se para uma próxima letra.



Figura 2.4 – Prática Virtual Celta - Treinamento Braille.

O aplicativo conta com três botões de controle, um para voltar ao desafio anterior, o outro para repetir o desafio e outro para passar para o próximo desafio. Além disso, os desafios são organizados nas seguintes classes: Explora, 1-0, aeiou, a-z e nas demais séries utilizadas para o aprendizado do braille que correspondem ao que foi instruído no item 2.1 deste capítulo.

Para ter acesso ao aplicativo, basta entrar no seguinte link:

[<https://scratch.mit.edu/projects/380045807/>](https://scratch.mit.edu/projects/380045807/).

Este aplicativo também é indicado para os membros da equipe de elaboração do Celta para que possam treinar a leitura em braille e, assim, se tornarem mais contextualizados com a leitura.

2.3 Ideias para uso com jogos educacionais

O Celta pode ser utilizado em atividades lúdicas para integrar os estudantes usuários ou não do braille. A seguir, são apresentadas algumas propostas.

2.3.1 Jogo descubra o caractere

Utilizando diversos Celtas, os estudantes devem competir no jogo, que consiste de um desafio proposto pelo professor, conforme a proposta de níveis de dificuldade apresentada no item 2.1 *Exercícios com o Celta*. O professor, sem que os alunos saibam, deve escrever uma sequência de caracteres e carregá-la no software do Celta. Depois, os alunos devem identificar quais são os caracteres. Quem terminar antes e com a maior quantidade de acerto é o vencedor.

Os desafios devem ser fornecidos de forma bastante gradual para que o jogo não se torne frustrante. Antes, pode ser realizado um aquecimento com a turma toda.

Os alunos videntes podem usar papel para escrever as respostas e os alunos cegos podem utilizar a reglete para este fim.

Como prêmio, pode se estabelecer algo construtivo para o crescimento da turma como um todo.

2.3.2 Descubra a palavra

Este jogo é semelhante ao anterior, porém agora será explorada a possibilidade de o Celta pular para a próxima palavra. São carregadas no software do Celta uma lista de palavras em uma ordem qualquer e os alunos terão que escrevê-las em um papel. Porém, antes de se iniciar o desafio, o professor já fala quais são as palavras, mas em ordem alfabética, assim os jogadores assim que identificarem os primeiros caracteres das palavras já podem pular para a próxima, possibilitando uma maior agilidade.

3. Como construir um Celta

Neste capítulo, fornecemos informações para a replicação do Celta. Incentivamos as instituições, incluindo escolas e outras organizações educacionais ou de apoio a pessoas com deficiência, que crie uma equipe multidisciplinar para se envolver na confecção e utilização do Celta. Se em sua instituição não existe pessoal técnico em algum aspecto, não há problema, pois sempre é possível buscar auxílio com a contratação de terceiros ou empresas para realizar parte a construção. Por exemplo: Se uma equipe não dispõe de impressora 3D, é possível terceirizar esta fase para alguma empresa. E assim por diante. Porém, é importante que a equipe, qualquer que seja ela, leia as instruções deste capítulo para ter uma visão global do processo.

A seguir, apresentamos a confecção da célula, a confecção do box, a confecção da placa de controle, a preparação dos Motores e a montagem final. Iniciemos com uma apresentação geral desse processo.

3.1 Apresentação

Como já informamos, o Celta é uma abreviação de Célula Tátil. O objetivo principal do Celta é ensinar, de maneira simples e dinâmica, a leitura do alfabeto braille. O hardware é composto por uma célula impressa em impressora 3D (vide a Figura 3.1, uma placa de Arduino UNO, uma placa de controle e um box). E o software é uma aplicação para *Microsoft Windows*, desenvolvida na linguagem em C++.



Figura 3.1 - Célula do Celta isolada. Fonte: Dos autores.

Para a célula e o box, nós fornecemos toda a modelagem em 3D, o projeto em CAD (*computer-aided design*), viabilizando a impressão em impressora 3D. Para a placa de controle, fornecemos todos os modelos necessários para a sua confecção, incluindo o diagrama elétrico da placa. Além disso, fornecemos todas as instruções para a confecção e montagem bem como a aplicação para *Microsoft Windows*.

O braille é um sistema de escrita tátil composto por um padrão de 6 pontos, ordenados em duas colunas e três linhas, numerados de 1 a 3 na coluna da esquerda e de 4 a 6 na coluna da direita, sempre de cima para baixo. No Celta estamos usando um padrão Internacional para a distância entre os pontos. Ou seja, a prática da sensibilidade realizada no Celta tende a ser semelhante a de outros displays braille comerciais.

Todo caractere alfanumérico pode ser representado em braille. Então, por exemplo, se o ponto 1 estiver levantado e os outros abaixados, tem-se o caractere “a” representado em braille; Se os pontos 1 e 2 estiverem levantados, tem-se a letra “b”; Se o 1 e o 4 estiverem levantados, a letra “c”; e assim por diante.

O software do Celta funciona da seguinte maneira. Com o hardware do Celta conectado via cabo USB no computador em que está instalado o Windows, copia-se um texto qualquer no computador e por meio dos botões do hardware do Celta este texto é inserido no software do Celta, permitindo sua leitura. O texto será lido caractere a caractere, sequencialmente, conforme o comando de avançar ou retroceder fornecido pelo usuário ao hardware. Se o usuário desejar pular para outra palavra, deve segurar o botão respectivo por dois segundos. Isso foi pensado porque, muitas vezes, a pessoa com deficiência visual já prevê a continuação de uma palavra graças ao contexto e deseja ir logo para a próxima.

Como este é um projeto aberto, já vislumbramos possibilidades de expansão do projeto pensando-o de forma modular ao colocar várias células uma ao lado da outra, similar a um display braille comercial. Para isso, será necessário apenas uma modificação no software e na placa eletrônica para tornar possível a leitura de vários caracteres ao mesmo tempo.

Para entender melhor sobre este tópico, assista ao vídeo disponibilizado no seguinte link:

<https://www.youtube.com/watch?v=29Zvcs4ppus>.

O vídeo é apresentado por Cristian Pendenza, que foi o principal responsável pela concepção tecnológica do projeto.

A seguir, prosseguiremos com a explanação de como confeccionar e montar um Celta.

3.1.1 Quais conhecimentos são necessários para a construção?

Do ponto de vista prático, as principais competências necessárias para a realização da confecção do Celta são as seguintes:

1. Entender o processo de modelagem em 3D.
2. Saber imprimir um arquivo dado em uma impressora 3D.
3. Entender, em linhas gerais, o circuito elétrico utilizado no Celta.
4. Entender, em linhas gerais, sobre a placa Arduino Uno.
5. Saber fazer soldagem de componentes eletrônicos.
6. Saber utilizar lima, lixa, chave de fenda, mini morsa etc.
7. Saber imprimir em impressora laser convencional.
8. Saber criar uma placa de circuito impresso (opcional).
9. Ter paciência e organização para lidar com peças pequenas.

3.1.2 Qual pessoal é necessário?

Com a Internet, atualmente qualquer pessoa interessada consegue aprender a realizar autonomamente todas as etapas de confecção do Celta, e este manual foi concebido para pessoas com este perfil. Porém, se algum gestor desejar focar e contratar alguém técnico capaz de executar facilmente toda esta construção, indica-se alguém com formação em mecatrônica ou áreas afins.

3.1.3 Que infraestrutura, equipamentos e insumos são necessários?

Para a construção completa de um Celta, indica-se uma sala bem iluminada com uma bancada de trabalho para eletrônica. Será necessária, ainda, uma Impressora 3D e um computador. Em termos de material permanente, indica-se o seguinte, conforme previsto no Quadro 3.1.

Quadro 3.1 – Material permanente para o projeto.

Quantidade e descrição do item	Valor aproximado
1 Retífica Dremel 3000 com 10 Acessórios, 110v	R\$ 260,00
1 Fonte de bancada 0 -15v 2A	R\$ 155,00
1 Jogo chaves fenda/phillips	R\$ 35,00
1 Multímetro digital convencional	R\$ 45,00
1 Mini morsa (torno de bancada)	R\$ 30,00
1 Paquímetro digital	R\$ 65,00
1 Ferro de soldar	R\$ 45,00
Valor total:	R\$590,00

Em termos de material de consumo, o projeto prevê a aquisição dos materiais listados nos Quadros 3.2 e 3.3, respectivamente para a bancada e para o circuito elétrico específico.

Quadro 3.2 – Material de consumo para a bancada.

Quantidade e descrição do item	Valor aproximado
1 Kit brocas precisão 0,8 A 3,2 Mm Ref. 628 Com 7 Peças Dremel	R\$ 38,00
1 Kit ferramentas para retífica Dremel uso geral	R\$ 117,00
1 Pistola de cola quente + refil	R\$ 20,00
1 Jogo chaves fenda/phillips	R\$ 35,00
1 Espaguete Termo Contrátil - 4 Bitolas (2, 3, 4 e 6mm x 5m)	R\$ 16,00
1 Kit de limas Agulha para acabamento	R\$ 43
2 Protoboard de 400 pontos	R\$ 20,00
1 Filamento ABS 1.75mm 1kg Branco	R\$ 85,00
1 Filamento ABS 1.75mm 1kg Preto	R\$ 85,00
2 Rolo de fita Isolante	R\$ 10,00
1 Rolo de Estanho para solda	R\$ 20,00
1 Rolo de fita crepe	R\$ 10,00
1 Kit para confecção de placa de circuito impresso ¹⁵	R\$ 160,00
Valor total:	R\$659,00

¹⁵ Incluindo: tinta fotossensível para circuito (azulada), tinta fotossensível para máscara verde, tinta fotossensível para simbologia branca, revelador de tinta fotossensível, removedor de tinta fotossensível, placa de fenolite de 100 x 100 mm, caixa plástica para corrosão, percloroeto de ferro em pó, transparência jato de tinta para fotolitos 1 folha, transparência a laser para fotolito 1 folha e lâmpada de 26w 110v negra uv.

Quadro 3.3 – Material de consumo para o circuito elétrico e a montagem final

Quantidade e descrição do item	Valor aproximado
3 Conectores DIP-1X2P-2.54MM	R\$ 14,00
2 Conectores DIP 2510S-12P	R\$ 10,00
3 Soquetes DIP-16	R\$ 9,00
2 Resistores de 330 ohms, 5%, 1/4W	R\$ 3,00
6 Motores de vibracall de celular (<i>DC Micro Vibration Motor Cell Phone Coreless Vibrator 4.4x4.4mm DC1.5-3V.</i>)	R\$ 90,00
2 Push bottons com 12mm de diâmetro para encaixe	R\$ 6,00
3 CIs L293D	R\$ 39,00
1 Arduino Uno R3	R\$ 30,00
3 Barras de 40 pinos Fêmea 180°	R\$ 9,00
1 Barra de soquete header	R\$ 1,00
5 Metros de cabos flexíveis 24 AWG, 0,20mm de cores variadas (incluindo verde, preto, vermelho e azul)	R\$ 5,00
1 Pacote com 40 Cabos flexíveis com conectores Dupont macho e fêmea	R\$ 15,00
1 Placa ilhada especial para Shield de Arduino Uno R3	R\$ 20,00
1 Placa folheada de cobre, dupla face para PCB	R\$ 10,00
2 Parafusos de 30mmX2mm com porca	R\$ 1,00
12 Parafusos de 7mmx2,5mm	R\$ 1,00
1 Fonte de alimentação DC 5V ,1A, positivo no centro e negativo por fora	R\$ 15,00
Valor total:	R\$278,00

O que podemos observar nesses quadros é que se já for disponibilizada uma oficina completa (representada pelos itens dos Quadros 3.1 e 3.2), o custo de cada Celta ficará próximo ao total do Quadro 3.3, que é o material para o circuito-elétrico. Os outros materiais de consumo serviriam para a montagem de diversos Celtas. De qualquer modo, se somarmos os 3 quadros e mais o valor de uma impressora 3D (de cerca de R\$2.000,00), uma instituição deveria gastar cerca de R\$3.500,00 com equipamentos e materiais para um Celta e depois mais cerca de R\$ 300,00 para fazer cada um dos Celtas seguintes.

3.1.4 Quais os softwares e sites que serão utilizados?

Em termos de softwares utilizados na explanação da construção do Celta, são importantes de serem considerados os seguintes:

- EasyEDA (opcional)
- Tinkercad
- IDE do Arduino
- Leitor de PDF
- Photoshop (opcional)
- Simplify3D (ou outro software fatiador, *licer*)
- Site do projeto: <https://Celta.ufscar.br>

Com exceção do *Photoshop* (que é opcional) e do *Simplify3D* (que pode ser substituído por outro) todos os softwares são de uso gratuito.

Com os materiais e equipamentos em mãos, podemos partir para a confecção, como veremos na sequência.

3.2 Confecção da célula

A célula é o principal elemento do Celta. Ela é construída de diversas peças plásticas (impressas em impressora 3D), 6 motores (como os utilizados em vibracall de telefones celulares), 12 cabos flexíveis (com conectores dupont macho em um dos lados) de cores variadas e dois parafusos de fixação. Vide a Figura 3.2.

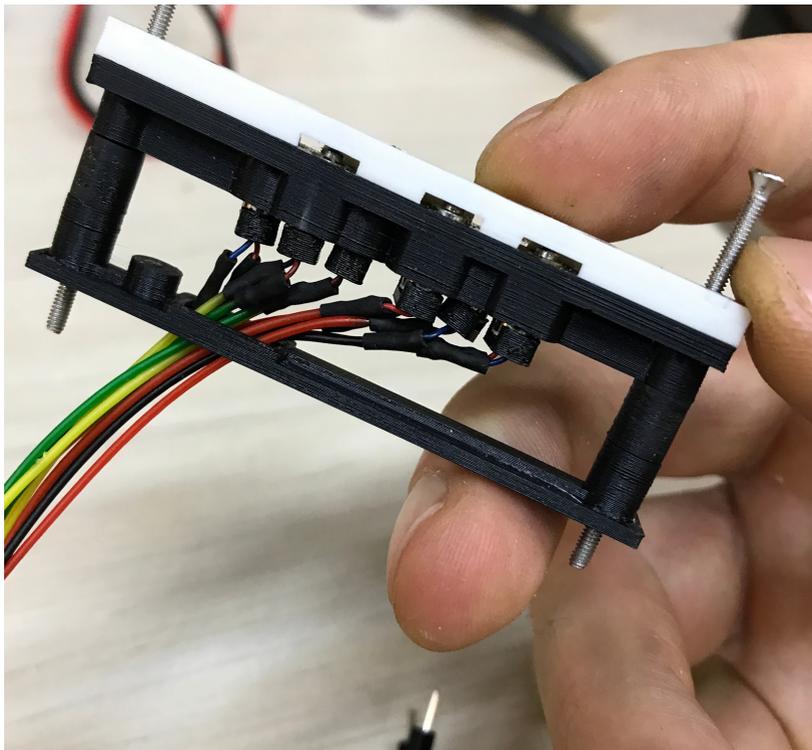


Figura 3.2 – Célula pronta vista pela lateral. Fonte: Dos autores.

A seguir, descreveremos como confeccionar e montar uma célula dessa.

3.2.1 Projeto CAD da célula

Para a impressão das diversas peças plásticas em impressora 3D, como já vimos anteriormente neste material, é necessária, inicialmente, a modelagem em 3D em um software, no caso um software CAD (*computer-aided design*). Em nosso projeto utilizamos o software *on-line* Tinkercad. Você vai precisar de uma conta no site <tinkercad.com>, é gratuito. Depois que você estiver logado em sua conta, entre no seguinte link que contém nosso projeto CAD (modelagem), chamado de *Celta_Cell_RELEASE-0.0.2*:

<<https://www.tinkercad.com/things/jgzAcNGjWxt>>.

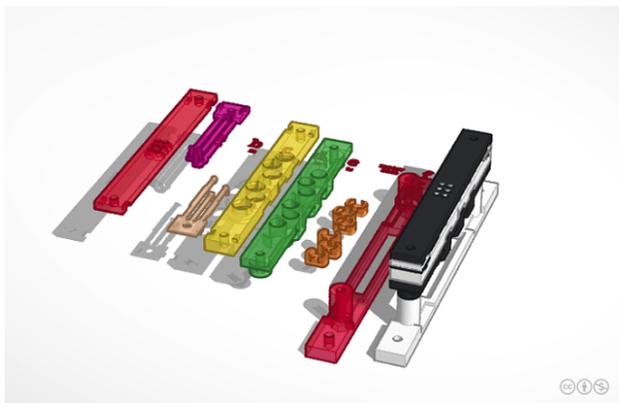
Depois de abrir o link (vide a Figura 3.2) você deve clicar em *Copiar e usar*.

Celta_Cell_RELEASE-0.0.2

##Display Braille

#Cel Brailit

♥ Curtir 0



projeto de:

Glauber Santiago

Box para o projeto Celta: Sistema de célula tátil para leitura Braille SEaD/UFSCar com apoio da CAPES. Site do projeto: <https://celta.ufscar.br>

Editado 4/8/20, Criado 4/6/20

Editar isto

Baixar

Thingiverse

Pin It



Incorporar

Visualizar em 3D

Adicionar imagem



Link: <https://www.tinkercad.com/things/jgzAcNGjWxt>

Figura 3.3 – Visão da página de apresentação com arquivo com a modelagem das peças da célula. Fonte: Dos autores.

A Figura 3.4 apresenta uma visão de cima dos diferentes tipos de peças modeladas (indicados pelas letras de “a” a “f”) e as respectivas quantidades a serem impressas.

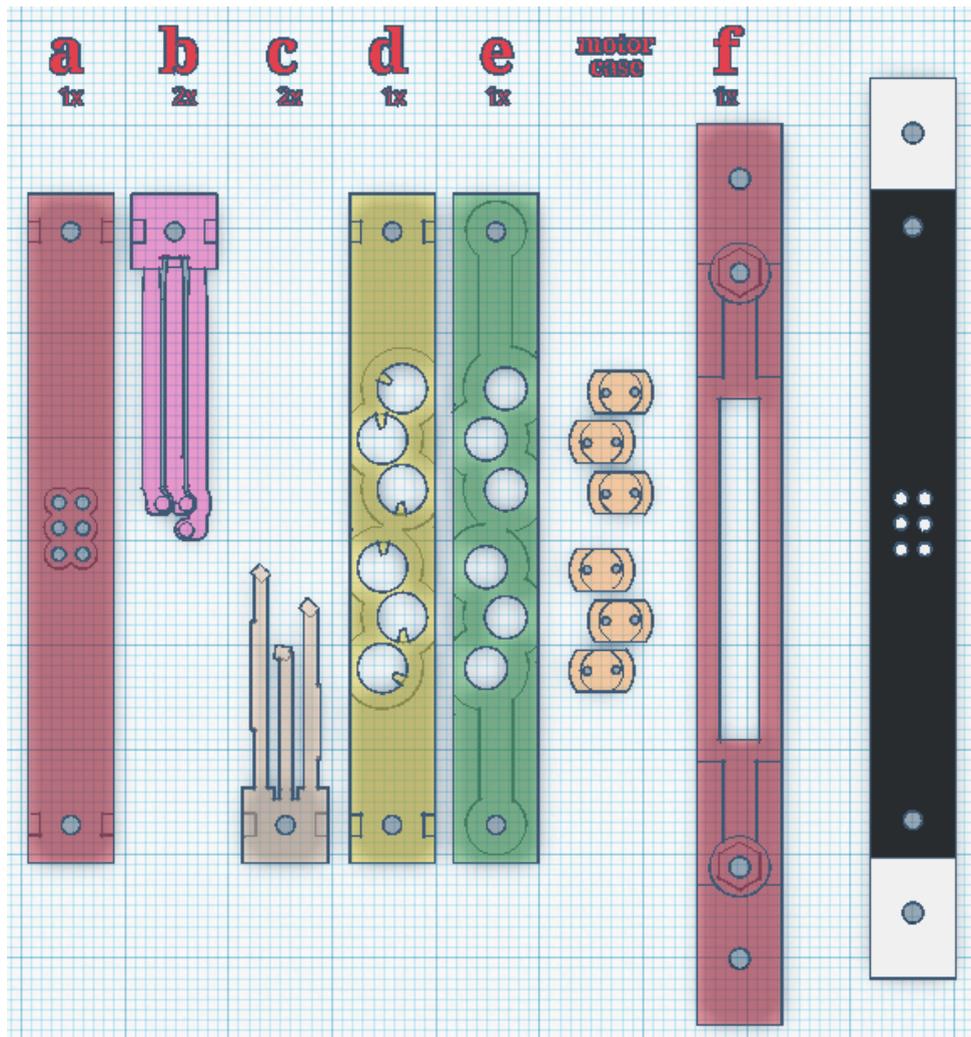


Figura 3.4 – Visão de cima das peças modeladas e as quantidades a serem impressas. Fonte: Dos autores.

As peças de “a” a “f” estão ordenadas de cima para baixo, conforme ilustra a Figura 3.5. Observe ainda que os suportes dos motores (“motor cases”) ficariam localizados entre as peças “e” e “f”. Porém, no momento da montagem, na verdade a primeira coisa que se encaixa são estes “cases” nos motores, pois são utilizados

para facilitar o manuseio dos motores e evitar que os fios conectores dos motores se danifiquem durante a montagem, pois são muito frágeis.

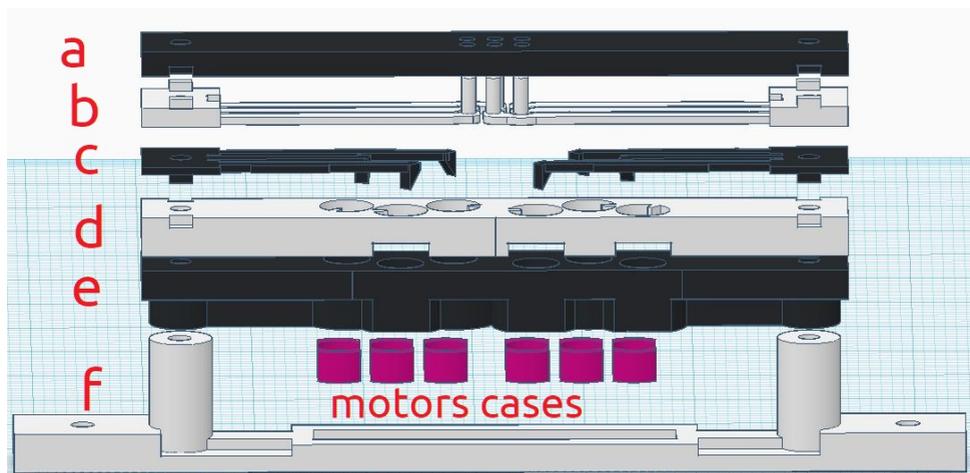


Figura 3.5 – Visão lateral com a ordem de montagem das peças. Fonte: Dos autores.

Para facilitar a organização das peças, utilizamos duas cores para impressão, como ilustrado na figura anterior. Você pode fazer isso também, conforme desejar, inclusive pode utilizar mais cores ainda. Isso dependerá das cores de material que você tiver ou das facilidades de troca de material de sua impressora. A impressora que utilizamos neste projeto foi uma *GTMax A1 Dual*, que possui a facilidade de suportar duas cores ao mesmo tempo, isso facilita este tipo de trabalho. Mas, como dissemos, isso não altera em nada a funcionalidade do projeto. A sequência de ordem de impressão das peças também se relaciona com o uso de variadas cores. Se sua impressora suporta apenas uma cor por vez, pode-se evitar trabalho se você imprimir todas as peças de uma cor e depois as peças da outra cor.

Seguindo a explanação do projeto CAD, temos, por exemplo, a peça “b”. Embora tenha apenas uma no projeto, elas serão duas. Você exportará apenas uma peça “b”, mas depois irá imprimir duas

vezes a mesma peça. Ao final, cada uma delas será utilizada em um dos lados internos da célula. Uma delas para os pontos 3, 4 e 5 e a outra para os pontos 2, 3 e 6 da célula braille (vide a Figura 3.6a).

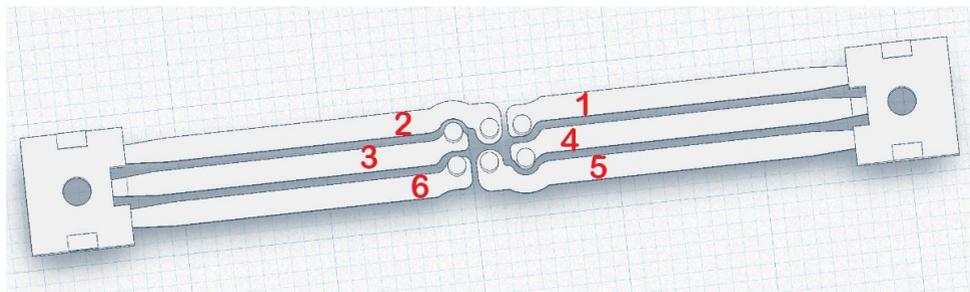


Figura 3.6a - Visão das duas cópias da peça “b” com a indicação dos pontos aos quais correspondem. Fonte: Dos autores.

A peças “c” também deverá ser impressa duas vezes. Já os “motores cases” podem ser exportados os 6 de uma vez. No geral, é recomendado que se imprima uma peça de cada vez. Mas, dependendo da peça, você pode imprimir duas, como no caso da “b” e da “c”.

Fora isso, existe o aspecto do acabamento pós impressão, que dependerá de sua impressora, mas talvez você tenha que dar um polimento com uma lixa fina ou com uma pequena lima em algumas peças. Isso foi necessário em nosso projeto piloto. Por mais que as configurações de impressão estejam ajustadas, é quase inevitável realizar pequenos ajustes.

Nesse sentido, um aspecto importante a ser considerado são os perfis de impressão. Ou seja, as configurações de impressão que você deve realizar no software de impressão da impressora 3D, também conhecido como fatiador. Por exemplo: As peças “a”, “b” e “c” possuem certas características anatômicas, e já testamos as configurações ideais de impressão para essas peças e as associamos a um perfil de impressão adequado a ser utilizado no fatiador. E assim por diante. Mas isso será discutido no momento oportuno.

Essa explanação anterior foi apenas para que se entenda a lógica das peças em nosso projeto de CAD. Por enquanto, o que importa é fazer os seguintes passos para exportar os arquivos modelados de cada peça:

1. Abra sua cópia do [Celta Cell RELEASE-0.0.2](https://www.tinkercad.com/things/jgzAcNGjWxt), conforme foi explicado anteriormente. O Link é o seguinte: <<https://www.tinkercad.com/things/jgzAcNGjWxt>>.
2. Para cada um dos tipos de peças (“a”, “b”, “c”, “d”, “e”, “f” e os “motores cases”), um de cada vez, clique na peça, depois clique no botão *Exportar* e depois clique na opção para baixar como *.STL*¹⁶, conforme ilustra a Figura 3.6b.

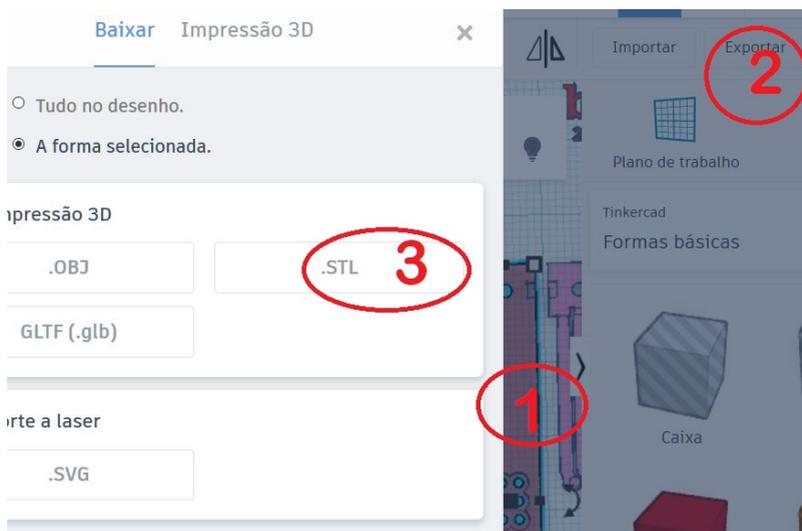


Figura 3.6b - Passos para exportar a arquivo stl de cada peça para impressão.

3. Salve os arquivos em seu computador com um nome correspondente à peça, para ficar fácil de você lembrar, por

¹⁶ *stl* é a abreviação de *stereolithography* (estereolitografia), que é um método de modelagem em 3D. O Arquivo *stl* é um arquivo com as informações de um objeto em 3D modelado.

exemplo “a”, “b” etc. Salve em um local separado, pois irá utilizá-los no momento da impressão. Aliás, são esses os arquivos que você enviaria para uma empresa de impressão em 3D, no caso de você não dispor de uma impressora.

Para entender melhor sobre este tópico, assista ao vídeo disponibilizado no seguinte link:

<<https://www.youtube.com/watch?v=LAN8gASXNF8>>.

3.2.2 Configurações do fatiador para a célula

O software que recebe as informações de um arquivo de modelagem 3D, como o arquivo *stl* exportado pelo Tinkercad, e transforma em um arquivo do tipo *G-Code*, é o arquivo com as informações necessárias para a impressora 3D imprimir o objeto em camadas. Esse software que gera o G-Code é chamado de fatiador, ou *Slicer*, ou o software que vem com a impressora, em um sentido mais informal. No fatiador, antes da impressão o que você deve buscar, em geral, é executar os seguintes passos:

1. Abra o seu software fatiador ([Netfabb Standard](#), [Cura](#), [PrusaSlicer](#), [OctoPrint](#), [MatterControl](#), [Simplify3D](#), [MakerBot Print](#) etc.). Geralmente, sua impressora já vem com um software como esses.
2. Importe (ou abra) o arquivo que você deseja imprimir, que no caso é um dos arquivos *.stl* que você exportou do Tinkercad, anteriormente.
3. Confira se o objeto está bem posicionado, no centro da mesa de impressão, segundo a visualização no fatiador.
4. Confira se o objeto está com a face mais adequada virada para baixo. Em nosso caso, quando for imprimir a tampa ela deve ser girada de cabeça para baixo para ficar com a parte plana para cima. Isso garantirá uma melhor impressão. Já o box em si, está na posição adequada.
5. Gere suportes de impressão para garantir uma boa qualidade em partes com orifícios ou partes “flutuantes”. Estes suportes

podem ser gerados automaticamente ou manualmente pelo software. Para o caso específico das peças da célula, estes suportes são dispensáveis.

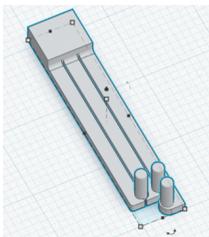
6. Escolha o preenchimento em 100%. Ou seja, no interior das peças não haverá espaços vazios. Isso melhora a durabilidade dos objetos, embora gaste mais filamento e aumente o tempo de impressão.
7. Faça testes para verificar qual a melhor temperatura a ser utilizada na impressão de cada objeto.
8. Faça testes para verificar qual a melhor velocidade a ser utilizada na impressão de cada objeto.
9. Confira a regulagem de altura da mesa em relação ao bico de impressão.

A Figura 3.7 apresenta um exemplo com algumas configurações que achamos ideais para as peças “a” e “b”.



A1.0 - Suporte micro pinos - parte superior

Material: ABS
Infill: 100%
Velocidade: 40m/s
Temperatura: 230 C



B1.5 - Haste suporte dos pinos com base

Material: ABS
Infill: 100%
Velocidade: 40m/s
Temperatura: 220 C

Observações gerais

B1.1

Velocidade impressão: 20m/s: muito lento, 80m/s muito rápido; 40mm/s por enquanto ideal;
Temperatura: 230C muito quente, 220C ideal;
Infill: 100%;
First layer: 200% (tornou as medidas exatas)

Figura 3.7 – Alguns parâmetros gerais de impressão.

No próximo tópico, iremos nos aprofundar nestes passos. Pelo momento, vamos focar em um outro aspecto.

Na elaboração inicial deste projeto, tínhamos disponível uma impressora *GTMx A1 Dual*, na qual o software fatiador *Simplify3D* já veio incluso. O que temos, a seguir, é uma explanação de como fizemos para chegar em uma melhor qualidade de impressão utilizando nossos recursos específicos. Pode ser bom para você entender esta experiência, como segue:

Configurações gerais com o Simplify3D para uma GTMax A1 Dual

Abra o software *Simplify3D* e o Arquivo com os perfis de impressão que disponibilizamos no seguinte link:

[<https://celta.ufscar.br/repositorio-de-arquivos/>](https://celta.ufscar.br/repositorio-de-arquivos/)

Então, vamos supor que você vai fatiar a peça “a” exportada pelo *Tinkercad*. Você deve adicionar a peça ao Simplify3D e ir em Select Profile e selecionar perfil de impressão chamado de Celta_Release-0.0._A. Vide a Figura 3.8. No item seguinte, iremos detalhar este processo. Por enquanto, vamos explorar as configurações gerais do Simplify3D para nosso projeto.

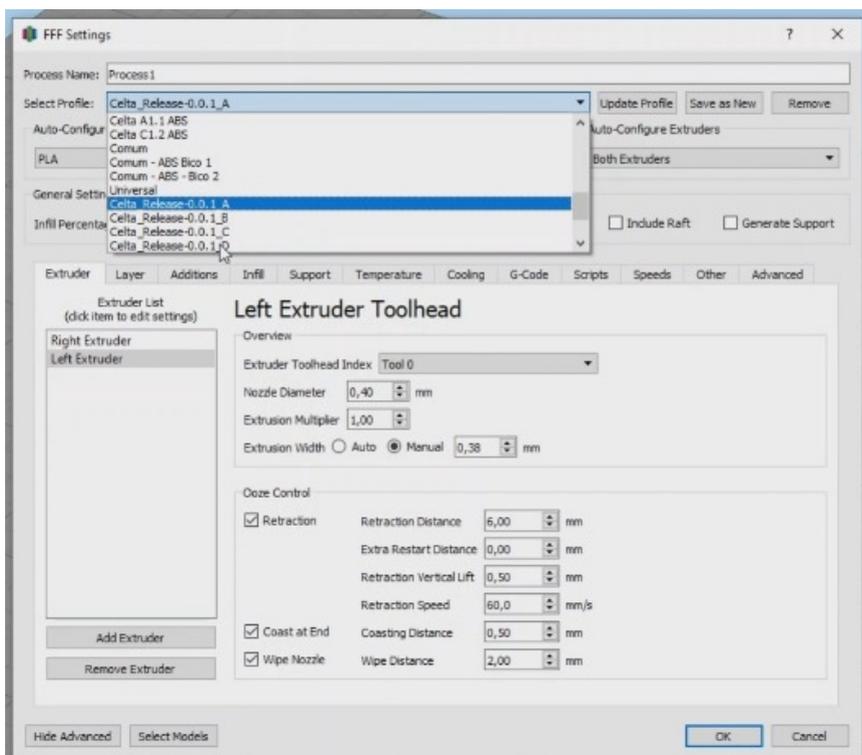


Figura 3.8 – Perfil A selecionado.

Depois, vá em *Extruder*, que é o bico por onde sai o filamento derretido. O bico (Extruder) é de geralmente de 0,4mm. Em nossa impressora, utilizamos 0,38mm para ficar mais fino. Mas não é um item obrigatório. Melhora pouca coisa! Vide a Figura 3.9.

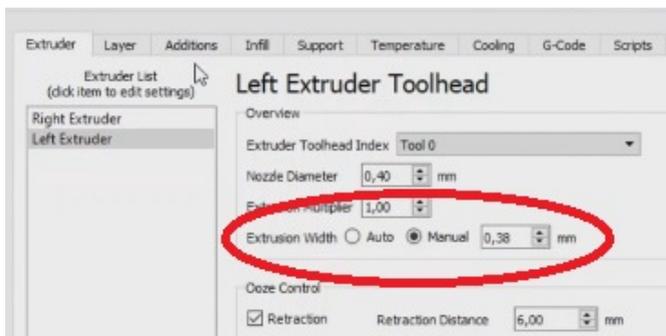


Figura 3.9 – Configuração do Extruder.

Depois, vá em *Layer* e verifique a altura de impressão do layer em *Primary Layer Height*. Aqui, colocamos este valor em 0,15mm. Já em *First Layer Height*, que é a altura da primeira camada, deixamos em 130mm. Como nossa impressora tem dois bicos, devemos indicar qual deles utilizar, então em *Primary Extruder* indicamos bico da esquerda (*Left Extruder*). Mas não são todas as impressoras que vão ter isso aqui. Não se preocupe! Vide a Figura 3.10.

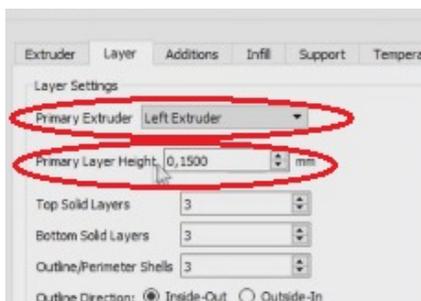


Figura 3.10 – Configuração do Layer.

Em *Additions*, indica-se usar o Skirt/Brim. Assim a impressora, antes de começar a imprimir de fato a peça, faz uma pequena borda em volta. Com isso, ela já vai soltando um pouco de filamento e quando a peça começa a ser impressa as coisas já estão funcionando corretamente. Vide a Figura 3.11.

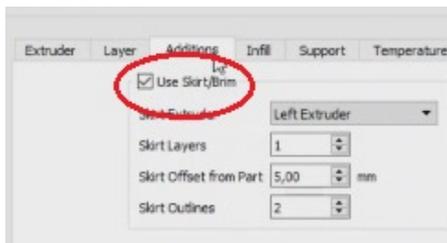


Figura 3.11 – Utilizando o Skirt/Brim

Em *Infill*, indica-se utilizar 100%. Em *Internal Fill Pattern*, escolha o tipo de *honeycomb* como *Full Honeycomb*. Este é o padrão que ela vai desenhar no preenchimento. Vide a Figura 3.12.

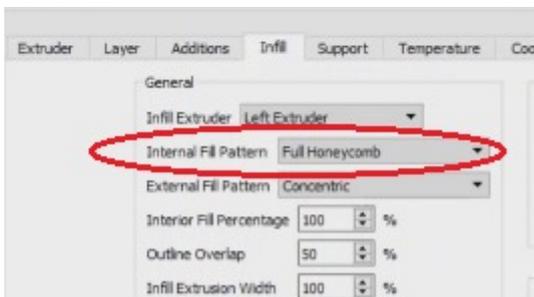


Figura 3.12 – Padrão de preenchimento.

Em *Support*, nenhuma alteração é necessária pois a anatomia dessas peças dispensam suportes.

Em *Temperature*, você deve ter mais cuidado. A bem da verdade, a temperatura ideal varia um pouco de filamento para filamento. Em nossa prática, tivemos 220°C como a ideal, embora o software recomende 230°C. Em *Heated Build Platform*, que é a temperatura da mesa, iniciamos 110°C. Como nossa impressora tem dois bicos, devemos indicar que o Primary Extruder é o bico da esquerda, Left Extruder. Vide a Figura 3.13.

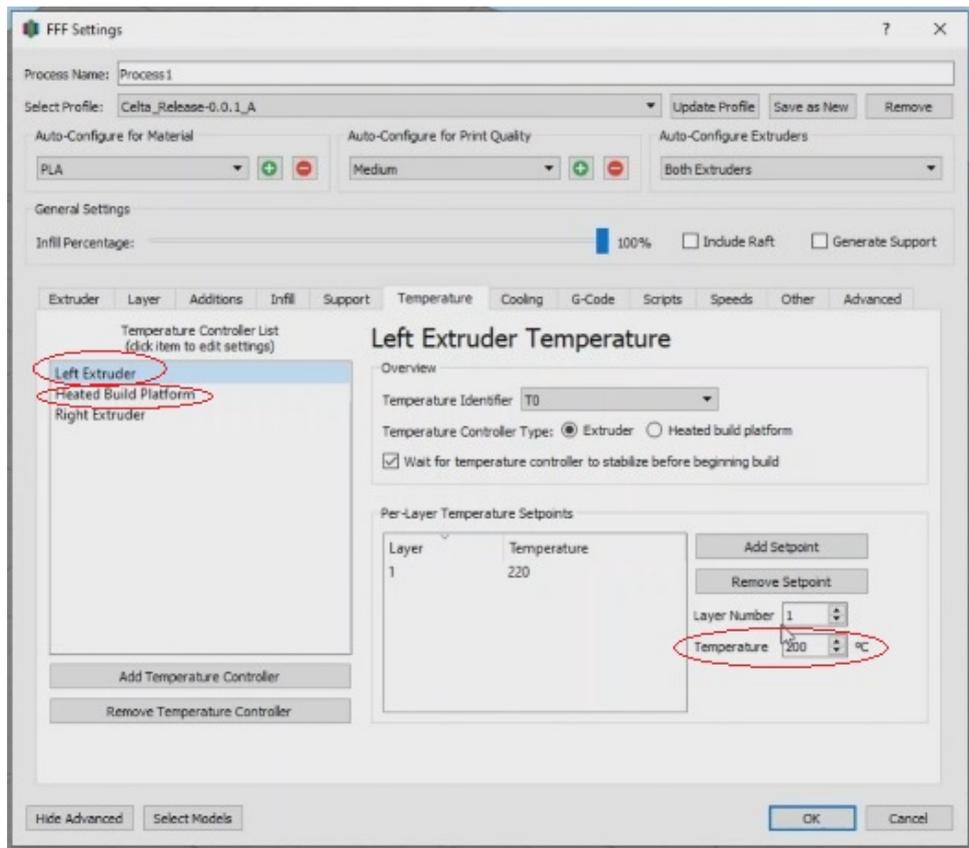


Figura 3.13 – Ajustes na temperatura.

Nossa impressora tem um pequeno ventilador, mas não o utilizamos no projeto, assim, em Cooling, deixe o ventilador desligado. Vide a Figura 3.14.

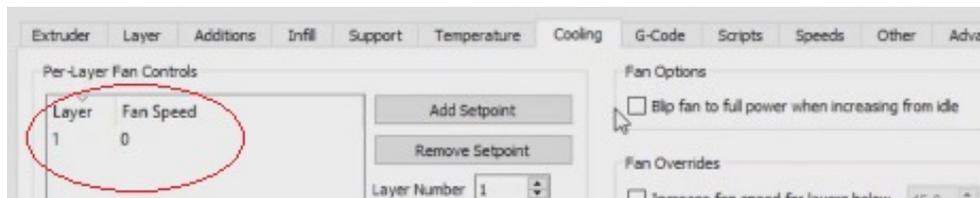


Figura 3.14 – Ventilador desligado.

Em *Speeds*, temos a velocidade que é algo bem importante. Tente deixar esse padrão de 20mm, conforme a Figura 3.15. Se tiver outro software, tente replicar estes padrões nele.

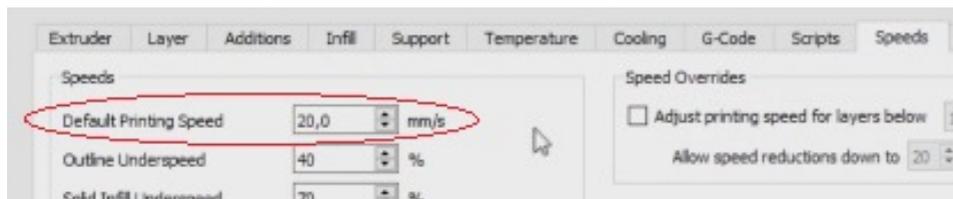


Figura 3.15 – Configurações de velocidade.

Um aspecto importante é a condição climática no interior da sala e da impressora, a impressora que utilizamos é fechada, vide a Figura 3.16.



Figura 3.16 – Impressora fechada.

Quando se imprime em ABS, a recomendação é utilizar uma impressora fechada para que a temperatura interior não diminua. Em uma impressora aberta o risco é maior. Por exemplo, quando se tem uma sala com ar condicionado este perigo é maior. O problema é que se a temperatura abaixar, pode ser que a peça comece a se desprender da mesa, danificando sua impressão. Isso ocorre principalmente com peças muito grandes, altas e que demoram a imprimir.

Se sua impressora for aberta, tente deligar o ar-condicionado e ainda construir uma proteção ao redor dela. Pode ser algo como uma caixa de papelão apenas em volta, próximo à mesa de impressão.

Para entender melhor sobre este tópico, assista ao vídeo disponibilizado no seguinte link:

<https://www.youtube.com/watch?v=LSDsrDxbaGs>.

3.2.3 Impressão da célula

Dando sequência aos procedimentos no fatiador para elaborar a impressão, você pode simplesmente realizar os 11 passos indicados inicialmente no item 3.2.2. Todavia, apresentamos, aqui, as possibilidades específicas que tivemos com a impressora e software fatiador (GTMax A1 Dual com o Simplify3D) quando da elaboração deste projeto. Vejamos:

Peças viradas para cima ou para baixo, em nossa experiência

Então, retomando o processo no Simplify3D, abra o software e abra novamente o arquivo com os perfis de impressão que disponibilizamos no seguinte link:

<https://celta.ufscar.br/repositorio-de-arquivos/>.

Depois, abra cada uma das peças exportadas pelo Tinkercad em formato stl. Inicie com a peça “a”. A peça aparecerá como na Figura 3.17.

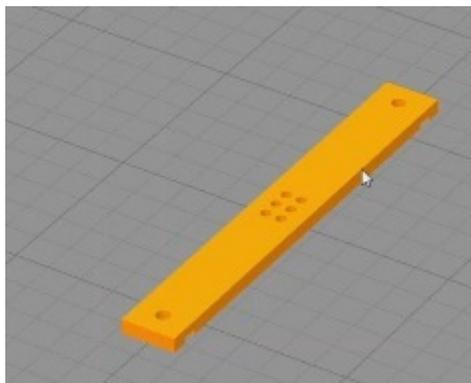


Figura 3.17 – Peça “a” vista de cima”.

Então, selecione o perfil de impressão correspondente. Ou seja, em *Select Profile* selecione o perfil de impressão chamado de *Celta_Release-0.0._A*.

Observando-se atentamente a peça “a” vemos que ela é lisa na parte de cima, mas na parte de baixo é perfurada. Veja a Figura 3.18 para notar esse detalhe, com ela virada em 180 graus.

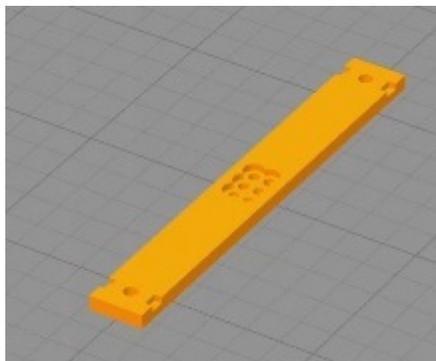


Figura 3.18 – Peça “a” vista de baixo”.

Ou seja, somente agora o lado liso está para baixo, que é como deveria ser.

Observe que o comando utilizado para girar a peça em 180° pode ser visualizado na Figura 3.19. Ou seja, a rotação do objeto em 180 no eixo y.

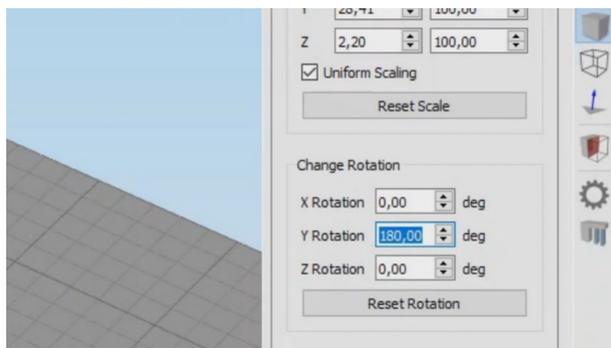


Figura 3.19 – Comando para rotacionar o objeto.

Sendo assim, a peça “a” está pronta para a impressão. Basta ir em *Prepare to Print* e iniciar a impressão, que pode ser via USB para a impressora ou via SD card, inserido diretamente na impressora. Utilizar o SD card é bom pois se der algum problema no computador, mesmo assim a impressão continua, já que não dependerá dele.

Existem mais algumas peças que devem ser rotacionadas, como a “c” (à qual chamamos de “scrips”). Vejam na Figura 3.20 como ela aparece. Está com as partes salientes para baixo e a parte reta para cima.

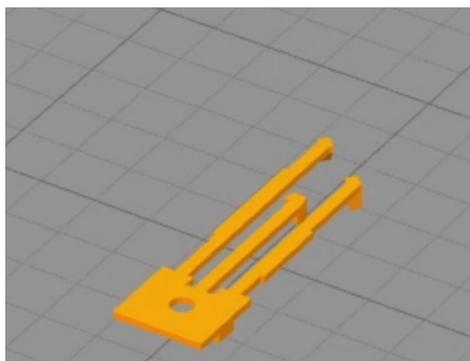


Figura 3.20 – Peça “c” como ver da modelagem.

Certamente deve ser rotacionada. Então ficará correta, como vemos na Figura 3.21.

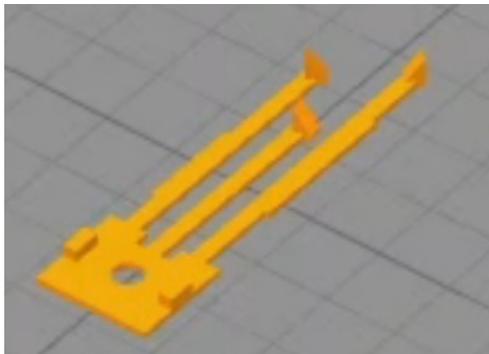


Figura 3.21 – Peça “c” rotacionada.

Então, agora, antes de imprimir a peça “c” basta selecionar o perfil adequado em *Select Profile: Celta_Release-0.0.1_C*.

Para a peça “d”, temos um pequeno detalhe. Ambos os lados possuem chanfros. Vide a Figura 3.22 que mostra a parte de baixo da peça.

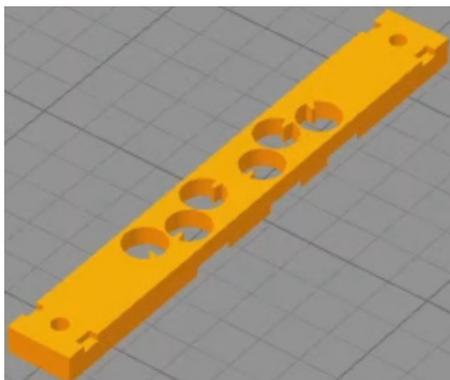


Figura 3.22 – peça “d” de cabeça para baixo”.

Porém, se observarmos a peça pelo lado de cima vemos que é bem mais complexa, conforme mostra a Figura 3.23.

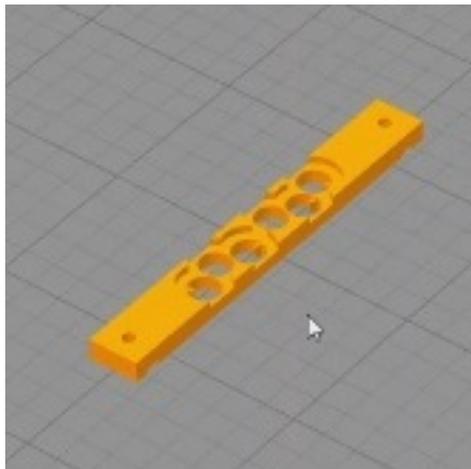


Figura 3.23 – peça “d” vista de cima”.

Bom, neste caso opta-se pelo menos ruim, que é deixar a peça como na Figura 3.24. É mais fácil perder um pouco de qualidade aqui nesse chanfro do que do outro lado. Aqueles dois vincos que ela tem é para encaixar na outra peça, então, no máximo será necessário aparar um pouco com um estilete. Na verdade, um pouco de acabamento manual é necessário fazer em quase todas as peças.

As demais peças não oferecem dificuldade, basta seguir estas ideias gerais.

Visualização das peças impressas

Para entender melhor os resultados do processo de impressão, veja as figuras seguintes e leia suas descrições.

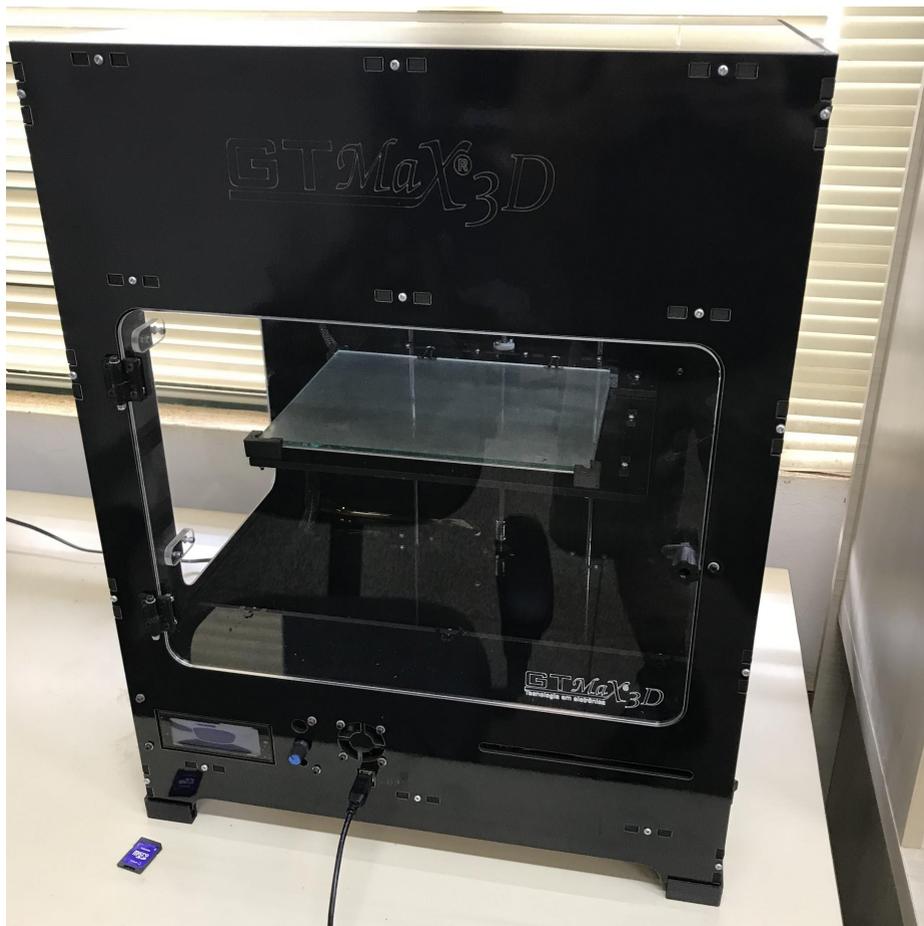


Figura 3.24 – Nossa impressora 3D, fechada, com cabo USB e com um SD card sendo utilizado.



Figura 3.25 – Bancada mostrando a necessidade de manter tudo organizado para não se perder com estas peças minúsculas.

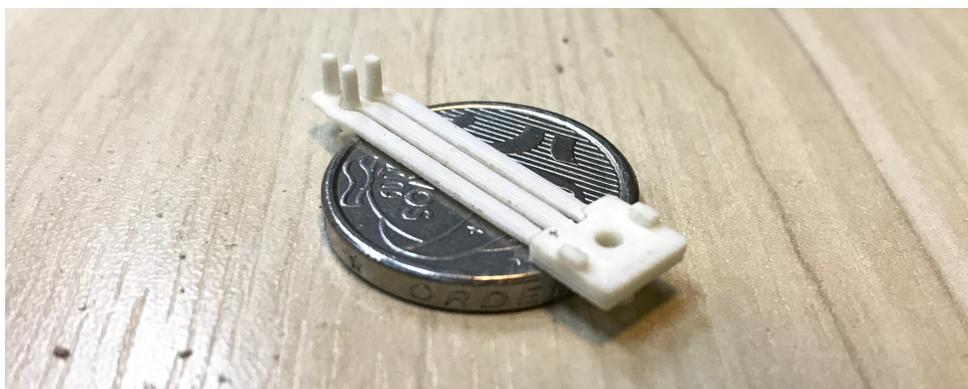


Figura 3.26 – Peça “b” impressa e sobre uma moeda de R\$ 0,50.



Figura 3.27 – Peça “c” impressa e sobre uma moeda de R\$ 0,50.



Figura 3.28 – Peça “d” impressa, ao lado de uma moeda de R\$ 0,50.



Figura 3.29 – Peça “e” impressa, apontada para uma moeda de R\$ 0,50.

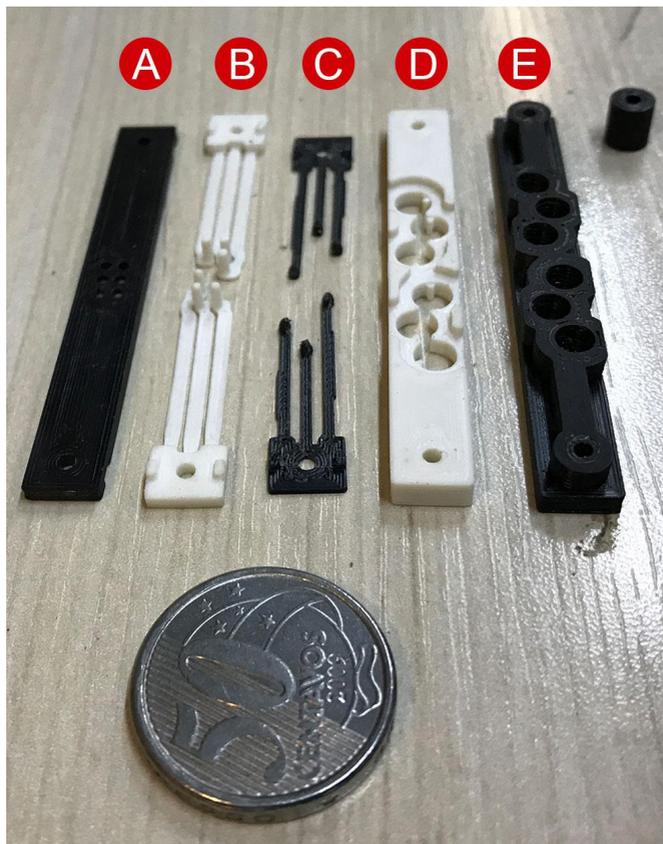


Figura 3.30 – Várias peças impressas, próximas a uma moeda de R\$ 0,50.



Figura 3.31 – Diversas peças impressas e sendo etiquetadas para facilitar o armazenamento e a montagem.

Para entender melhor sobre este tópico, assista ao vídeo disponibilizado no seguinte link:

<https://www.youtube.com/watch?v=r3T2u6w8BYM>.

3.3 Confecção do box

Além da célula, outro elemento do Celta é sua caixa. Neste tópico, aprenderemos a elaborar esta parte, mas será bem mais simples de entender, já que o processo é bastante semelhante ao que foi demonstrado na confecção da célula. Vamos lá!

3.3.1 Projeto CAD do box

Da mesma forma que apresentamos a modelagem no Tinkercad das peças plásticas da célula (no item 3.2.1), agora iremos comentar sobre o estojo, a caixa (box) na qual será envolvido o dispositivo.

Novamente entre em sua conta no [tinkercad.com](https://www.tinkercad.com). Depois abra o link do projeto chamado de Celta_Box_RELEASE-0.0.2:

<https://www.tinkercad.com/things/iValvIHpQ6w>.

Clique em Copiar e usar. Vide a Figura 3.32 para entender melhor.

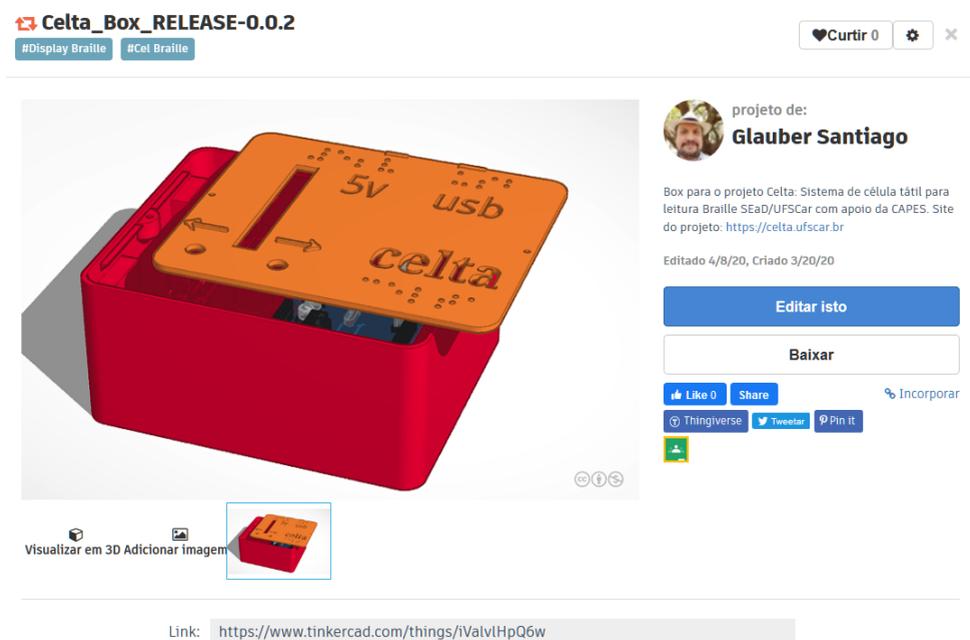


Figura 3.32 - Visão da página de apresentação com arquivo com a modelagem do box.

Ao abrir o projeto, ele apresentará os elementos como na Figura 3.33.

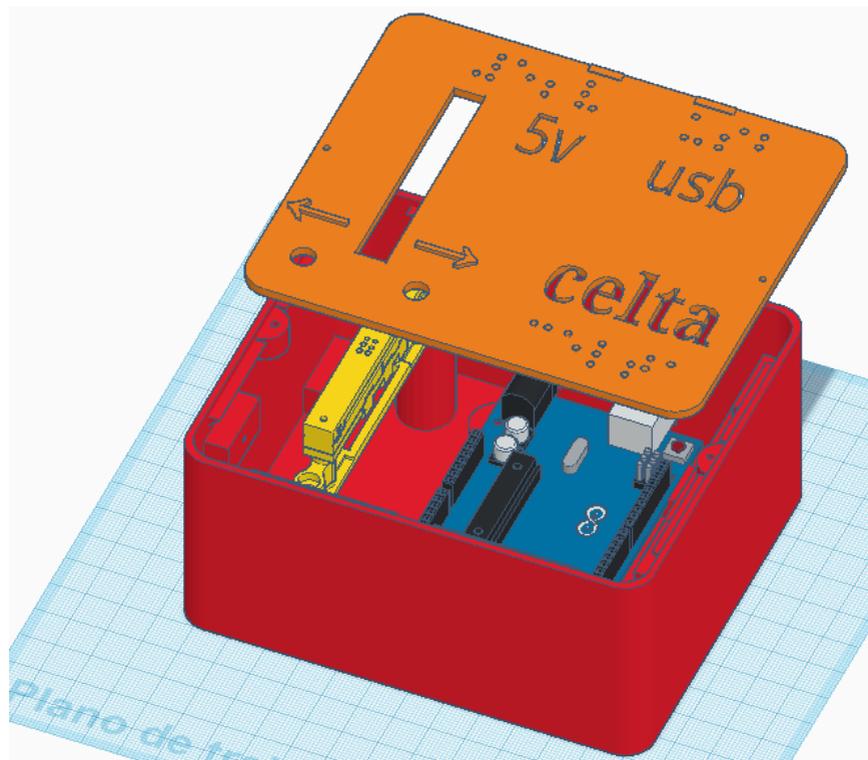


Figura 3.33 – Elementos disposto no projeto do Box.

Como vemos na figura, existem elementos do box e outros que são apenas ilustrações o Arduino e da Célula. Ou seja, estes elementos devem ser descartados na impressão. Deve-se imprimir apenas o box (vermelho) e sua tampa (alaranjada). O box possui 130x110mmX60mm. Vide a Figura 3.34 para visualizar o box e sua tampa, aberta.

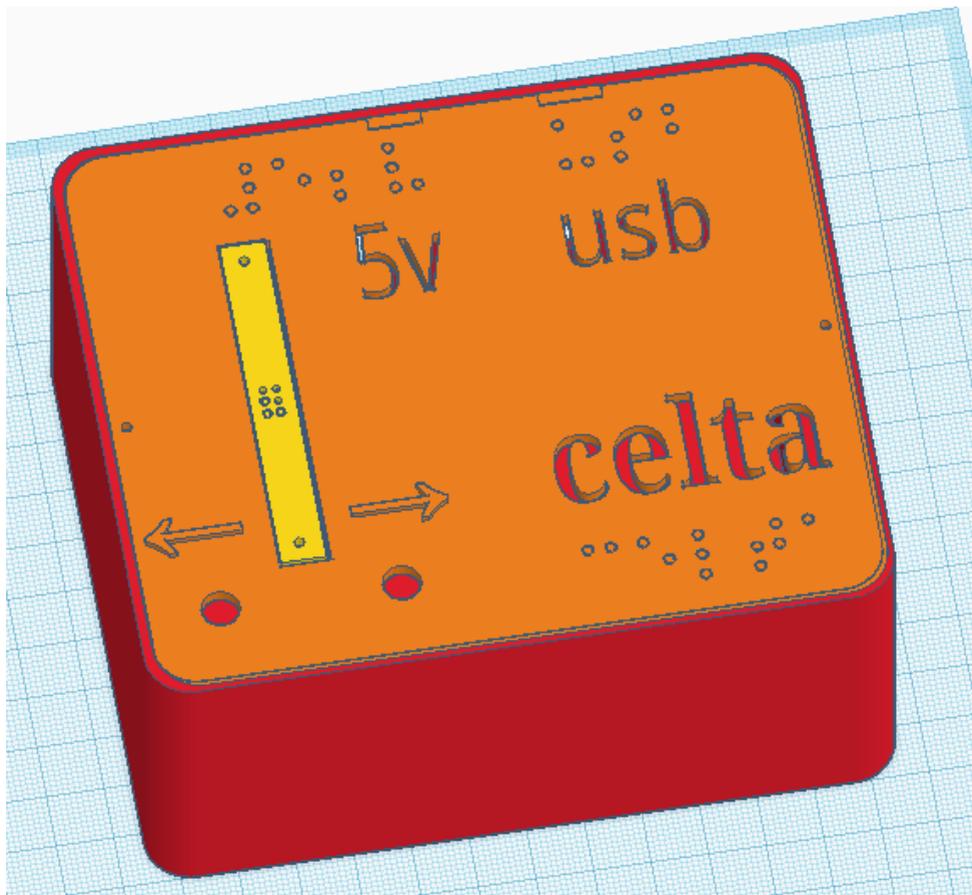


Figura 3.34 – Box com seu conteúdo e tampa fechada.

Como dissemos, além do box e sua tampa o projeto CAD inclui as demais peças e o Arduino. Elas são apenas para você se guiar, para saber como irá montar, depois, toda a célula. Por enquanto, é necessário que você oculte tudo o que não será utilizado. Para isso, basta clicar nos elementos desnecessários e apertar o botão ocultar, como ilustra a Figura 3.35, na qual foi selecionado o Arduino e depois está visível o botão de ocultar, circulado em amarelo.



Figura 3.35 – Ocultando um objeto selecionado.

Ao final, é para ficar visível apenas o seguinte, como mostra a Figura 3.36.

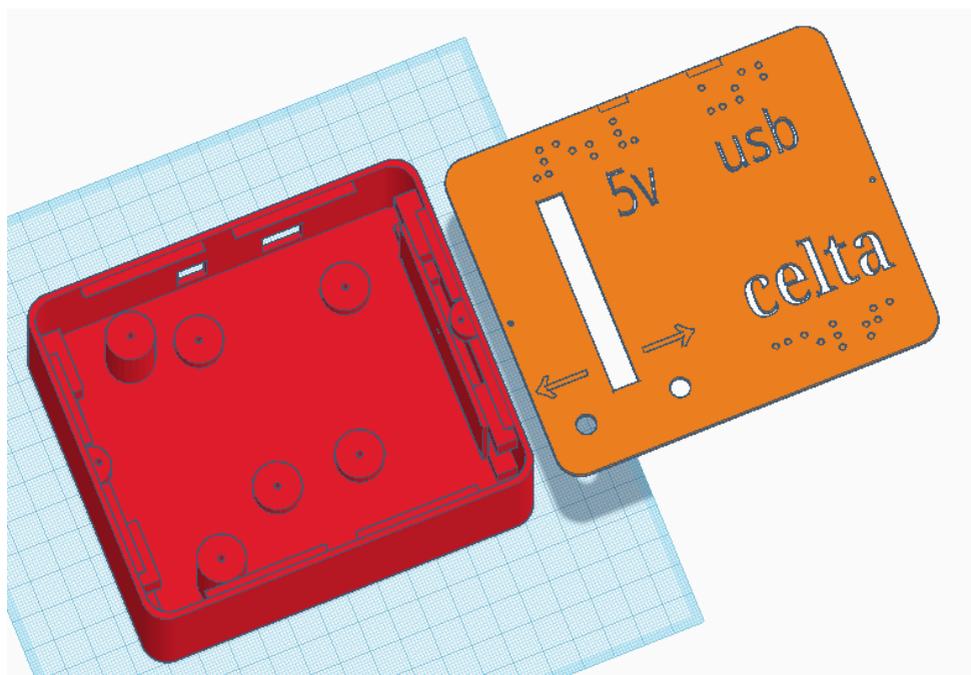


Figura 3.36 – Elementos finais que devem ficar visíveis no CAD do box.

Agora, basta que você salve os dois objetos modelados restantes, um de cada vez. Para isso, clique no objeto, depois clique no botão *Exportar* e depois clique na opção para baixar como *.STL*.

Deixe esses dois arquivos guardados, pois serão eles os arquivos a serem utilizados pelo software da impressora 3D.

Para entender melhor sobre este tópico, assista ao vídeo disponibilizado no seguinte link:

<<https://www.youtube.com/watch?v=WsJhkZZW78w>>.

Para visualizar um exemplo de impressão assista ao seguinte vídeo que elaboramos na impressão de um protótipo inicial do box. Vídeo - *Impressão e retirada de suporte do protótipo*:

< <https://youtu.be/wmgU2HlmxTE>>.

Disponibilizamos ainda o vídeo *Retirada dos suportes do box* que mostra o box em sua versão final e com seus suportes sendo extraídos após a impressão. O link é o seguinte:

<<https://youtu.be/Yx9LkpKjvsA>>.

3.3.2 Configurações do fatiador e de impressão para o box

Para imprimir os dois arquivos stl que você salvou no item anterior, relativos ao box e sua tampa, você deverá utilizar o software específico que vem com sua impressora. A grande questão é que imprimir objetos em impressora 3D requer uma certa quantidade de paciência e experiência de uso com cada modelo e marca de impressora. Porém, em linhas gerais, indica-se os mesmos passos em qualquer software fatiador, que tratamos anteriormente no caso da célula:

1. Abra o seu software fatiador ([Netfabb Standard](#), [Cura](#), [PrusaSlicer](#), [OctoPrint](#), [MatterControl](#), [Simplify3D](#), [MakerBot Print](#) etc.). Geralmente, sua impressora já vem com um software como esses.
2. Importe (ou abra) o arquivo que você deseja imprimir.

3. Confira se o objeto está bem posicionado, no centro da mesa de impressão, segundo a visualização no fatiador.
4. Confira se o objeto está com a face mais adequada virada para baixo. Em nosso caso, quando for imprimir a tampa ela deve ser girada de cabeça para baixo para ficar com a parte plana para cima. Isso garantirá uma melhor impressão. Já o box em si, está na posição adequada.
5. Gere suportes de impressão para garantir uma boa qualidade em partes com orifícios ou partes “flutuantes”. Estes suportes podem ser gerados automaticamente ou manualmente pelo software.
6. Escolha o preenchimento em 100%. Ou seja, no interior das peças não haverá espaços vazios. Isso melhora a durabilidade dos objetos, embora gaste mais filamento e aumente o tempo de impressão.
7. Faça testes para verificar qual a melhor temperatura a ser utilizada na impressão de cada objeto.
8. Faça testes para verificar qual a melhor velocidade a ser utilizada na impressão de cada objeto.
9. Confira a regulagem de altura da mesa em relação ao bico de impressão.

Nós tínhamos uma impressora GTMax A1 Dual, que já veio com o software Simplify3D. Então, já disponibilizamos aqui todo nosso know how para o caso de você também possuir este software e esta impressora.

Fatiando box no Simplify3D

Logo que abrimos o arquivo stl do box no Simplify3D, temos a visão semelhante à que aparece na Figura 3.37. Observe que esta parte deste capítulo foi elaborada com as figuras de um protótipo inicial do box, por isso você pode notar algumas diferenças na peça. Mas de resto o processo é o mesmo.

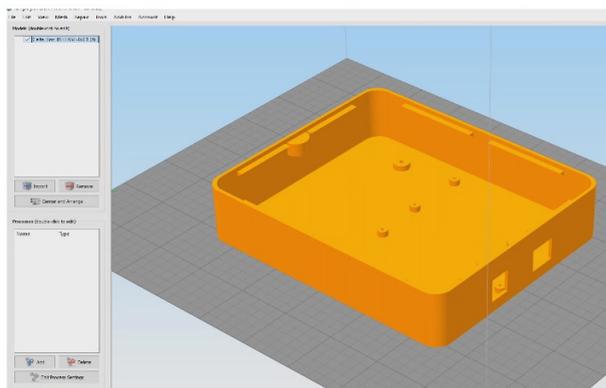


Figura 3.37 – Simplify3D: visão do box.

Então, o primeiro passo é adicionar os suportes para garantir uma melhor impressão. Para isso, clique em *Support Generation* e depois em *Generate Automatic Supports*. A Figura 3.38 mostra como fica com os suportes no modo automático.

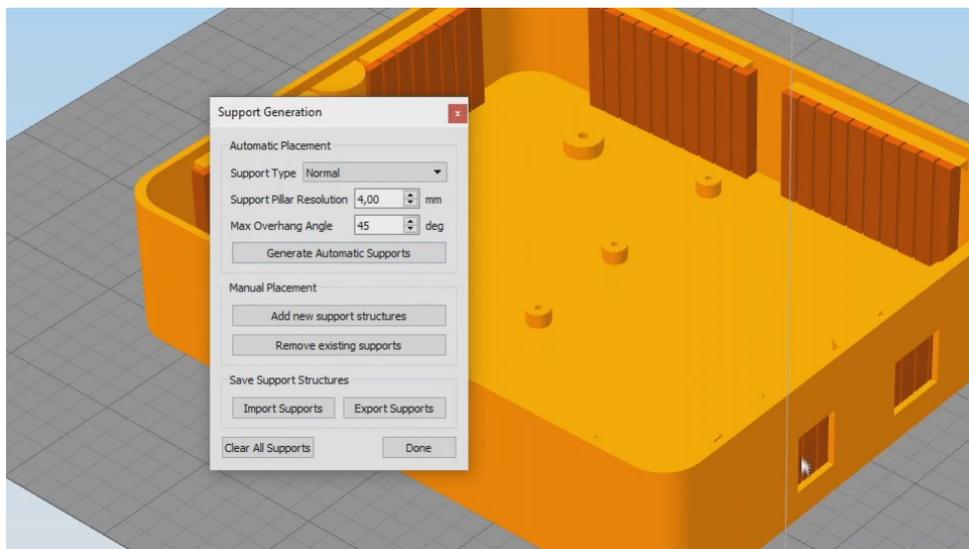


Figura 3.38 – Simplify3D: com suportes automáticos.

Porém, em nossa experiência vimos que no automático alguns suportes são desnecessários para o nosso box, então, optamos por

apagar os suportes para depois os criarmos manualmente. Para isso, clicamos em *Clear All Supports*. Vide a Figura 3.39.

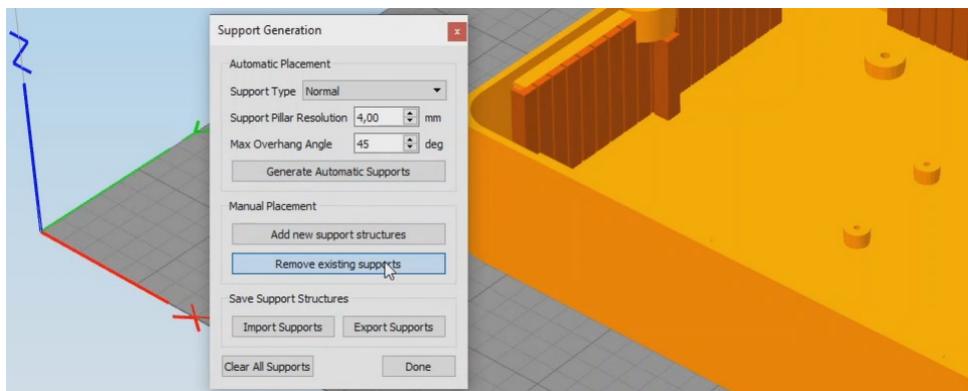


Figura 3.39 – Simplify3D: Removendo os suportes.

Depois clique no botão *Add new support structures*, como mostra a Figura 3.40.

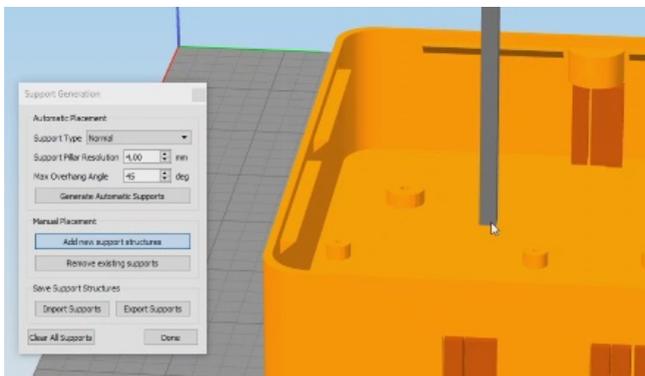


Figura 3.40 – Simplify3D: adicionando os suportes manuais.

Vá adicionando um a um, para ficar tal qual a Figura 3.41.



Figura 3.41 – Simplify3D: visão com os suportes manuais suficientes.

Para os usuários do Simplify3D, também já disponibilizamos juntamente com os arquivos do projeto os arquivos com as configurações de impressão. São perfis de impressão. Para o box, utilize o seguinte perfil (profile): *Celta_Release-0.0.1 Box-Case*. Depois, clique em *Save as New* e no botão *Ok* e estará tudo pronto para imprimir o box. Vide a Figura 3.42.

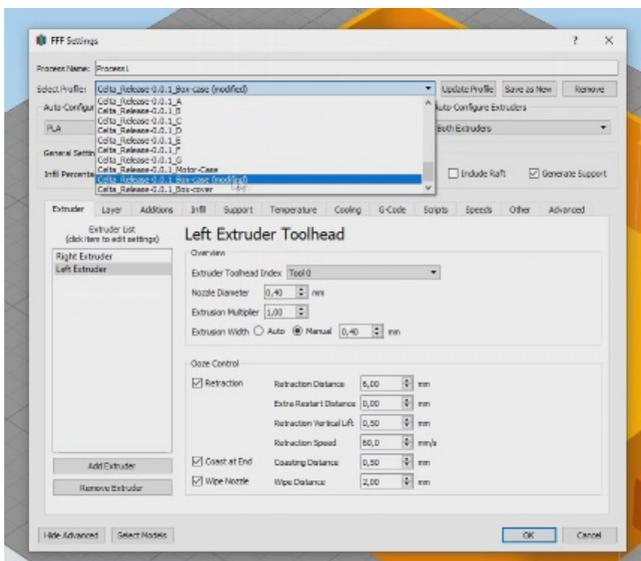


Figura 3.42 – Simplify3D: Seleção do perfil de impressão para o box.

Para a impressão da tampa do box, basta abrir o arquivo stl respectivo. Ficará como está na Figura 3.43.

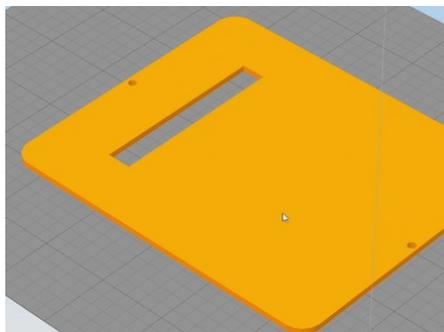


Figura 3.43 – Simplify3D: visão do box.

Para esta tampa não existe necessidade de ajustes de suportes, basta utilizar o perfil de impressão que já disponibilizamos, conforme indica a Figura 3.44. Utilize o perfil (profile) *Celta_Release-0.0.1 Box-Cover*.

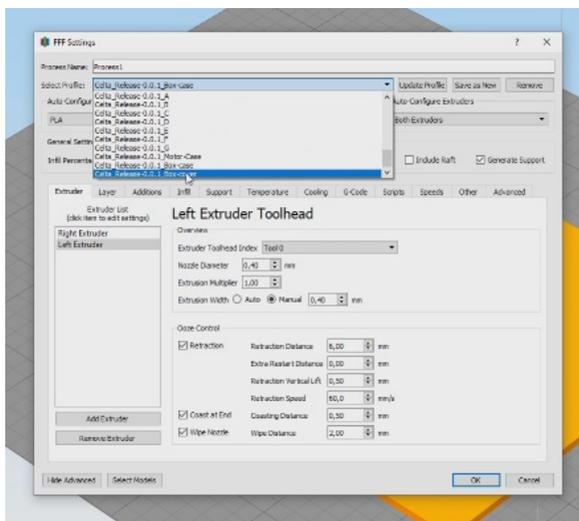


Figura 3.44 - Simplify3D: Seleção do perfil de impressão para a tampa.

Na Figura 3.45, podemos observar que existe a opção de impressão ser enviada diretamente do computador para a

impressora via cabo usb ou com um arquivo transferido para um cartão de memória. Para impressões mais demoradas, mais de uma hora por exemplo, indicamos que se utilize o envio para o cartão e este cartão ser inserido na impressora, pois, ocasionalmente, variações da corrente elétrica podem afetar o computador e este perder momentaneamente o contato com a impressora, ocasionando perda do material.

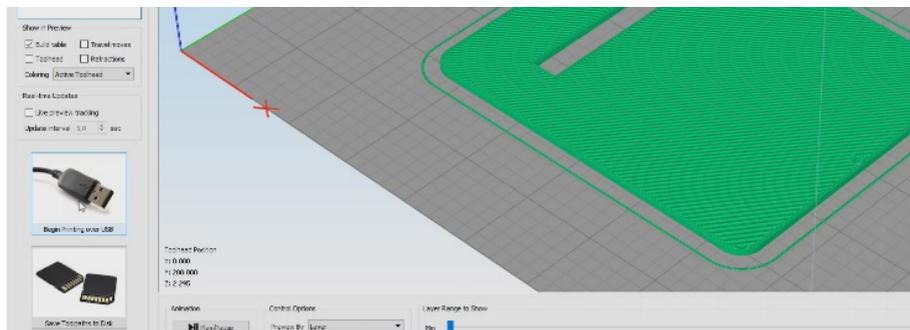


Figura 3.45 – Simplify3D: visão do box.

Para entender melhor sobre este tópico, assista ao vídeo disponibilizado no seguinte link:

<<https://www.youtube.com/watch?v=8GBzCLLChCI>>.

3.4 Confecção da placa de controle

Para além da célula, outra parte fundamental do Celta é sua placa de controle. Trata-se de uma placa de circuito impresso onde são realizadas as ligações dos componentes eletrônicos (resistores, CIs), da placa do Arduino, dos motores, dos botões e da fonte de alimentação.

Neste item, veremos as instruções necessárias para a elaboração desta placa.

3.4.1 Componentes eletrônicos necessários

Os componentes necessários para a confecção da placa estão listados no Quadro 3.4 a seguir. O quadro apresenta, novamente, a quantidade e a descrição de cada componente. Além disso, apresenta uma previsão do custo aproximado para a aquisição do componente. Indica-se a aquisição destes elementos em uma maior quantidade se o projeto for realizado por pessoas mais inexperientes. Isso evita contratemplos por conta de queima acidental de componentes eletrônicos. Mas, de qualquer forma, estes materiais totalizam cerca de R\$ 250,00 no mercado brasileiro atual.

Quadro 3.4 – Componentes da placa e seus custos.

Componentes necessários	Custo¹⁷ unitário	Custo total
3 Conectores DIP-1X2P-2.54MM	R\$ 7,00	R\$ 14,00
2 Conector DIP 2510S-12P	R\$ 5,00	R\$ 10,00
3 Soquete DIP-16	R\$ 3,00	R\$ 9,00
2 Resistores de 330 ohms, 5%, 1/4W	R\$ 1,00	R\$ 3,00
6 Motores de vibracall de celular (<i>DC Micro Vibration Motor Cell Phone Coreless Vibrator 4.4x4.4mm DC1.5-3V.</i>)	R\$ 15,00	R\$ 90,00
2 Push bottons com 12mm de diâmetro para encaixe	R\$ 3,00	R\$ 6,00
3 CIs L293D	R\$ 13,00	R\$ 39,00
3 Barras de 40 pinos Fêmea 180°	R\$ 3,00	R\$ 9,00
1 Barra de soquete header	R\$ 1,00	R\$ 1,00
1 Arduino Uno R3	R\$ 30,00	R\$ 30,00
5 Metros de cabos flexíveis 24 AWG, 0,20mm de cores variadas (incluindo verde, preto, vermelho e azul)	R\$ 1,00	R\$ 5,00
Pacote com 40 Cabos flexíveis com conectores Dupont macho e fêmea	R\$ 15,00	R\$ 15,00
1 Placa ilhada especial para Shield de Arduino Uno R3	R\$ 20,00	R\$ 20,00
1 Placa folheada de cobre, dupla face para PCB	R\$ 10,00	R\$ 10,00
Custo total dos componentes:		R\$261,00

¹⁷ Custo calculado em valores aproximados em março de 2020 no mercado brasileiro convencional. No caso da aquisição pela internet, este valor pode cair ao menos pela metade.

Após adquirir esses componentes, é possível iniciar a montagem. A seguir, vamos entender melhor esse aspecto.

3.4.2 Projeto e diagrama elétrico

Para entendermos o projeto em sua totalidade, é necessário um pouco de conhecimento sobre o diagrama (ou esquema) elétrico, conforme ilustrado na Figura 3.46.

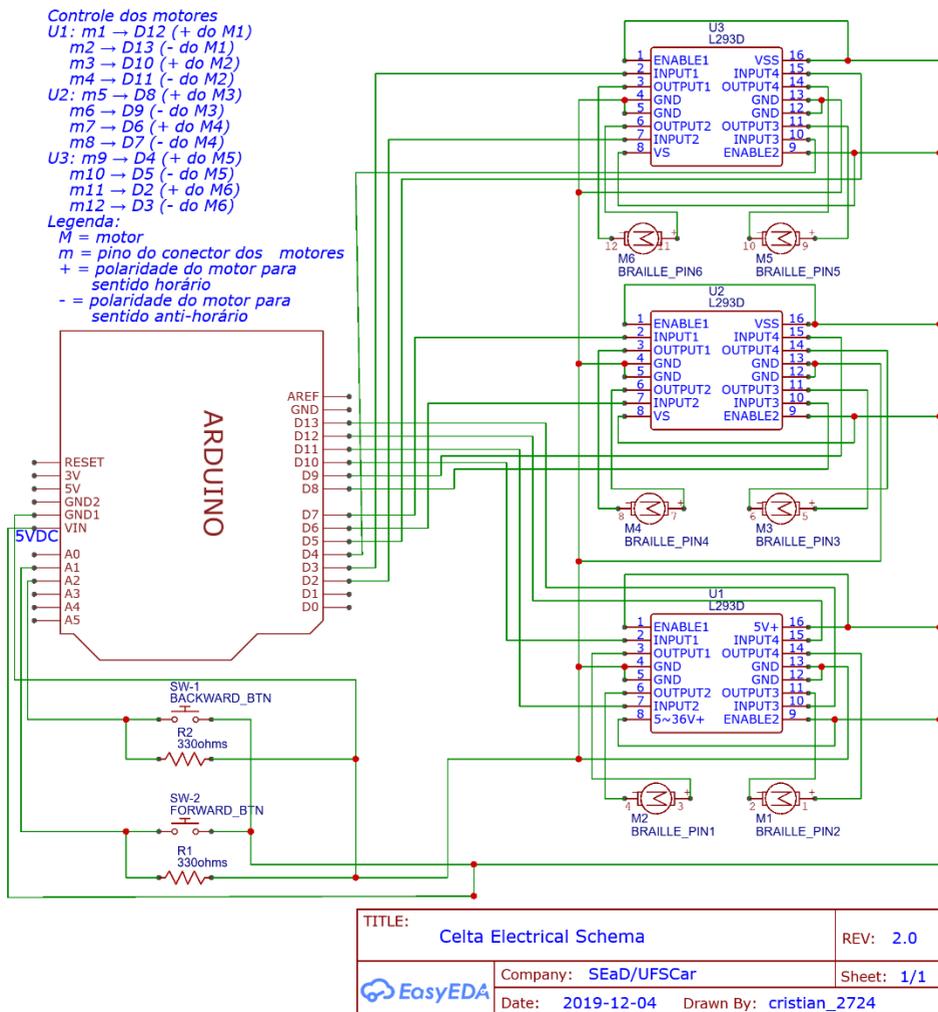


Figura 3.46 – Diagrama elétrico completo do Celta.

Em um esquema elétrico, as linhas (em verde) representam as conexões elétricas, os pontos representam que as linhas estão conectadas eletricamente e os diversos símbolos representam os componentes eletrônicos ou os conectores. O Quadro 3.5 apresenta um resumo da simbologia e dos componentes utilizados no esquema elétrico e na placa de controle.

Quadro 3.5 – Sumário da simbologia dos componentes utilizados no esquema elétrico.

Descrição e Legenda na Placa	Simbologia no diagrama	Fotografia
Resistor de 330 ohms, 5%, 1/4W (R1 e R2)		
Motores de vibracall de celular		
Push bottons (BWD_BTN e FWD_BTN)		

<p>CI L293D</p>		
<p>Arduino Uno R3</p>		

A lógica de funcionamento do Celta se dá pelo controle de 6 pequenos motores, DC. Ou seja, são motores que dada a polaridade elétrica em seus terminais eles giram em sentido horário ou em sentido anti-horário. Por exemplo, consideremos o motor 1 (M1). Ele está conectada ao circuito pelos conectores m1 e m2. O m1 está indicado como + e m2 está indicado como -. Porém, para estes motores, na realidade não quer dizer que o + sempre é positivo e o - sempre negativo. O fato é que quando o motor for alimentado com o + como positivo e o - com o negativo, nestas condições ele girará em sentido horário (fazendo com que o pino correspondente no Celta suba). E, quando ocorrer o oposto, ele girará em sentido anti-horário (fazendo com que o pino abaixo).

A parte principal do circuito são os circuitos integrados (CI) L293D. Eles são os responsáveis por controlar os motores. A

Figura 3.47 apresenta o detalhe do diagrama esquemático com a pinagem deste CI.

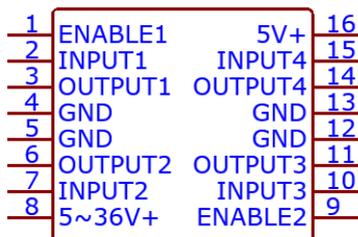


Figura 3.47 – Esquema externo do circuito integrado L293D.

Os pinos do CI são configurados da seguinte forma:

- 1-Enable1. Ativa os canais 1 e 2 do CI.
- 2-Input1. É a entrada de controle 1.
- 3-Output1. É a saída para um dos terminais do motor 1.
- 4-GND. É o *Ground* (terra).
- 5-GND. É o *Ground* (terra).
- 6-Output2. É a saída para o outro terminal do motor 1.
- 7-Input2. É a entrada de controle 2.
- 8-5~36V+, ou V_s (*Supply Voltage*). É a alimentação para os motores 1 e 2.
- 9-Enable2. Ativa os canais 3 e 4 do CI.
- 10-Input3. É a entrada de controle 3.
- 11-Output3. É a saída para um dos terminais do motor 2.
- 12-GND. É o *Ground* (terra).
- 13-GND. É o *Ground* (terra).
- 14-Output4. É a saída para o outro terminal do motor 2.
- 15-Input4. É a entrada de controle 4.
- 16-5V+, ou V_{ss} (*Logic Supply Voltage*). É a alimentação para a parte lógica do CI.

O CI L293 possui 4 entradas (Input), que são ligadas em portas digitais do Arduino, e 4 saídas (output), que são ligadas em cada um dos dois terminais de dois motores. Então, se o Arduino fornecer 5V pelas suas portas digitais à Entrada 1 e não fornecer nada para a Entrada 2 o motor 1 girará em um sentido. E se no lugar da Entrada 1, a Entrada 2 que for energizada, o motor 1 girará no sentido contrário. A Figura 3.48 ilustra uma montagem de um circuito simulando um único motor ligado ao CI L293 e controlado pelo Arduino. Para ter acesso a esta simulação, entre no link:

<https://www.tinkercad.com/things/3wRF7YdbZxw>.

No alto da Figura 3.48 vemos um código de programação utilizado para esta simulação. Ou seja, o sketch inicia com o pino 6 (porta 6) do Arduino em Alto (ou seja, 5V) e o pino 7 em Baixo (ou seja, 0V). Então, isso fará com que o motor gire no sentido horário. Depois de 1 segundo, o código indica o contrário, com o pino 6 em baixo e o 7 em alto, e por isso o motor girará em sentido anti-horário.

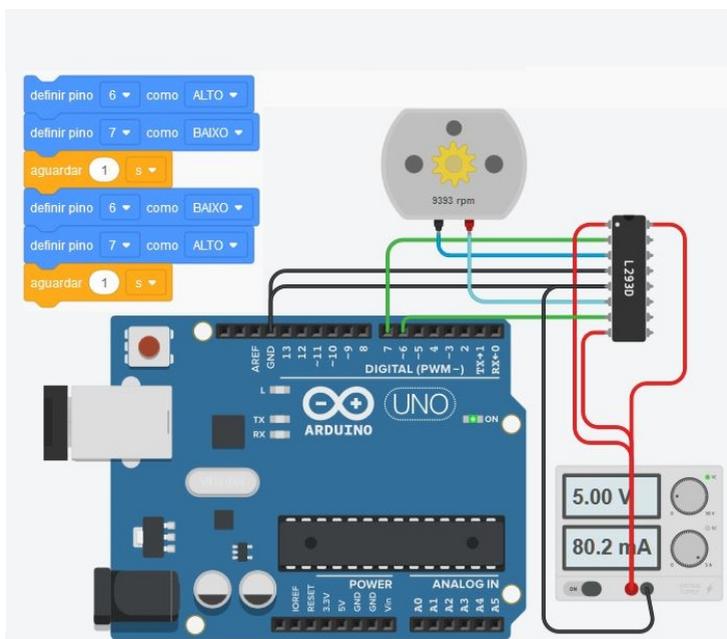


Figura 3.48 – Simulação com L293D com um motor no Tinkercad.

Na figura, ainda podemos observar as ligações básicas que são os pinos do CI referentes à alimentação do motor, à alimentação da parte lógica do CI e à habilitação dos canais 1 e 2 todos interligados e ligados à uma fonte de alimentação de 5V (veja a fiação em vermelho). Além disso, temos os pinos 4, 5, 12 e 13 que estão ligados ao terra (ver os fios de cor preta). Podemos ainda notar que a porta 6 do Arduino está ligada ao pino 2, que é a entrada 1 do CI e também a porta 7 do Arduino ligada ao pino 7 do CI que é a entrada 2 (veja os fios de cor verde). Por fim, temos em azul a fiação dos terminais do motor 1, ligados aos pinos de saída 1 e 2 do CI (pinos 3 e 6). Para se entender adequadamente como os 3 CIs estão conectados e quais são as portas digitais do Arduino que controlam cada motor, o seguinte esquema foi utilizado, conforme o Quadro 3.6.

Quadro 3.6 – Esquema das ligações dos motores e portas do Arduino.

CI	Motor	Conexão para motores	Terminal do motor	Porta digital de controle
U1	M1 (motor para o ponto 1 da célula)	m1	+	D12
		m2	-	D13
	M2 (motor para o ponto 2 da célula)	m3	+	D10
		m4	-	D11
U2	M3 (motor para o ponto 3 da célula)	m5	+	D8
		m6	-	D9
	M4 (motor para o ponto 4 da célula)	m7	+	D6
		m8	-	D7
U3	M5 (motor para o ponto 5 da célula)	m9	+	D4
		m10	-	D5
	M6 (motor para o ponto 6 da célula)	m11	+	D2
		m12	-	D3

Para se ter uma ideia de como é a ligação de todos os 6 motores, montamos a seguinte simulação, novamente, no Tinkercad, conforme a Figura 3.49a. O link para esta simulação é o seguinte:

<<https://www.tinkercad.com/things/6iX4XzukzZ2>>

Antes de montar seu Celta, indicamos que entenda bem estas ligações e lógicas de funcionamento. E estas simulações são interessantes para este processo.

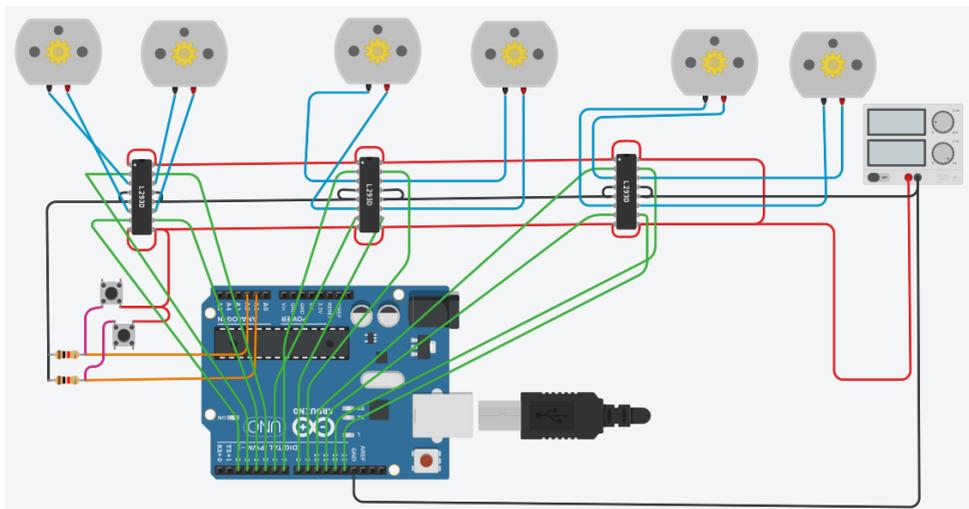


Figura 3.49a – Simulação com os 6 motores ligados.

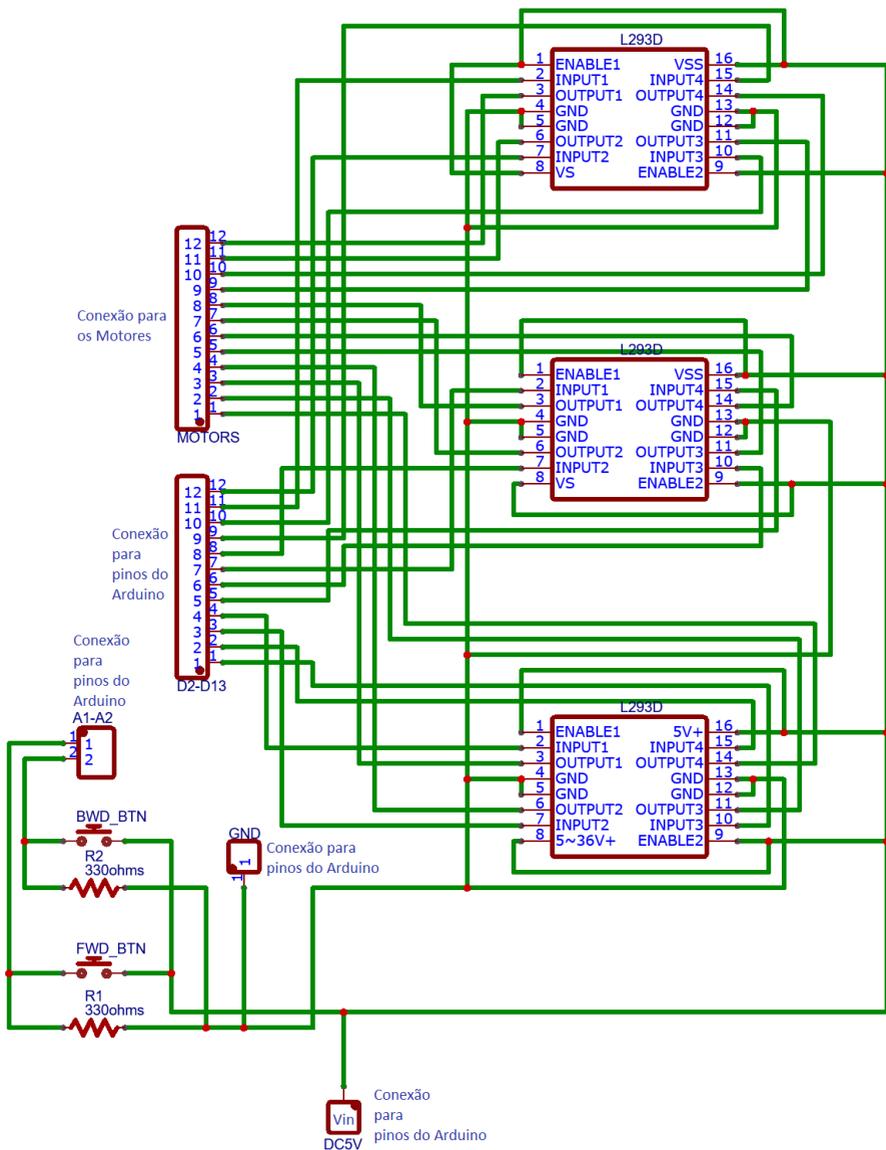
O Diagrama elétrico completo do Celta que vimos na Figura 3.46 é importante para se entender o funcionamento do circuito. Mas para a montagem da placa em si elaboramos um circuito simplificado¹⁸ que pode ser visualizado na Figura 3.49b. Neste diagrama, no lugar dos motores e do Arduino vemos simplesmente os conectores a serem utilizados.

¹⁸ Para a elaboração desses dois esquemas elétricos, utilizamos o software on-line EasyEDA¹⁸. Com ele, podemos desenhar um esquema elétrico, modificá-lo e ainda gerar o desenho utilizado para confeccionar a placa de circuito em si.

A seguir, temos o link para o projeto elétrico completo:

https://easyeda.com/cristian_2724/celta_electrical_schematic e também temos o link para o projeto simplificado, ou seja, apenas para a confecção da placa de controle:

https://easyeda.com/cristian_2724/celta_pcb_simplified



TITLE: Celta Schematic For PCB Production		REV: 1.0
	Company: SEaD	Sheet: 1/1
	Date: 2019-12-09	Drawn By: cristian_2724

Figura 3.49b – Esquema elétrico simplificado.

Como foi indicado anteriormente, temos a placa de controle (encaixada) conectada sobre o Arduino. Isso será feito por meio dos dois package DIP de doze (pelo MOTORS e pelo D2-D13). Além disso, na placa temos um conector padrão de cinco volts (DC5V). Temos ainda dois conectores (FWD_BTN e BWD_BTN) para a conexão dos botões de avançar e de voltar. Temos as indicações de R1 e R2 para os dois resistores e, ainda, GND para indicar o terra (neutro). Por fim, temos A1-A2 que são as conexões que vão ser ligadas às portas A1 e A2 do Arduino.

3.4.3 Montando uma placa ilhada

Em nosso projeto, concebemos duas opções para você montar a placa de circuito impresso de controle. Uma delas é utilizando uma placa ilhada e outra é confeccionando uma placa específica. Vamos focar agora na primeira abordagem que pode ser mais fácil de ser implementada.

Existem diversos tipos de placas ilhadas, mas recomendamos que se use uma placa ilhada especial para Shield de Arduino Uno R3, como a apresentada nas Figuras 3.50 e 3.51. Esta placa permite um encaixe perfeito sobre o Arduino, o que deixa a montagem final mais compacta e resistente.

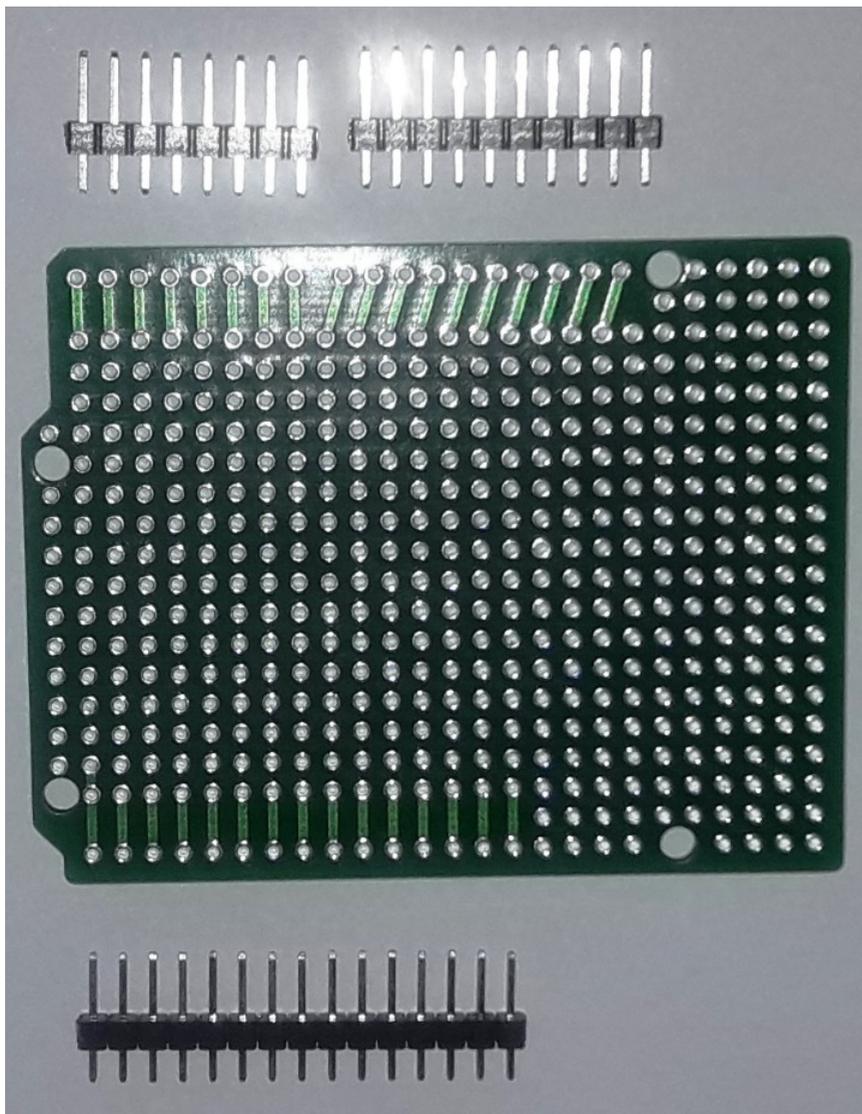


Figura 3.50 – Foto de Placailhada especial para Shield de Arduino Uno R3 incluindo terminais. Fonte: Dos autores.

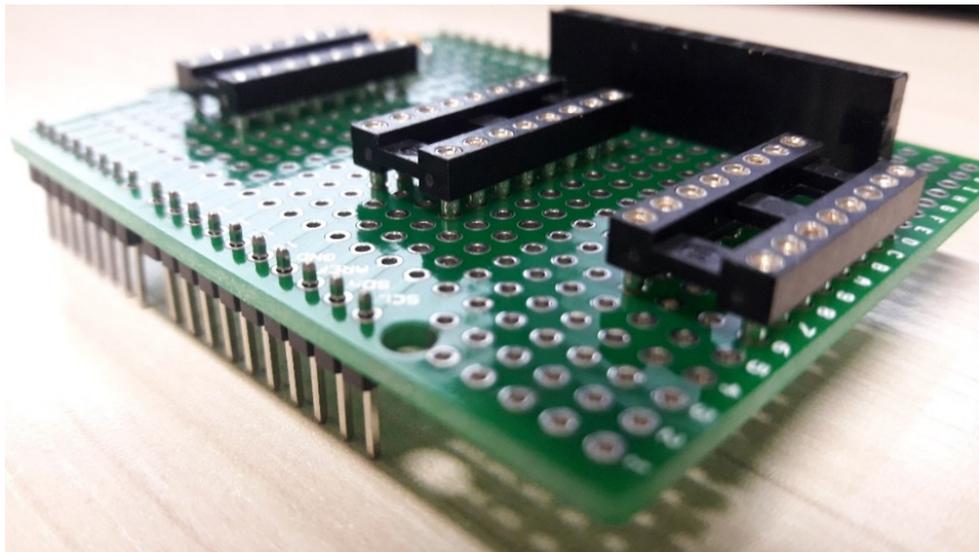


Figura 3.51 – Foto de Placa ilhada especial para Shield de Arduino Uno R3 incluindo soquetes dos CIs e terminais para os motores.

Fonte: Dos autores.

A lógica de montagem da placa ilhada é que se solde os componentes e depois cada uma das ilhas. Assim que as ilhas estiverem mais frias, se solda o espaço entre as ilhas de forma que eles se unam, conforme as necessidades do circuito, formando assim, trilhas. Para ligações que não foram possíveis por meio dessas trilhas, são utilizados fios.

Para facilitar a montagem da placa ilhada, elaboramos um esquema de montagem com a visão de frente e a visão espelhada da placa, para possibilitar um guia na soldagem de ambos os lados da placa. Vide as Figuras 3.52 e 3.53.

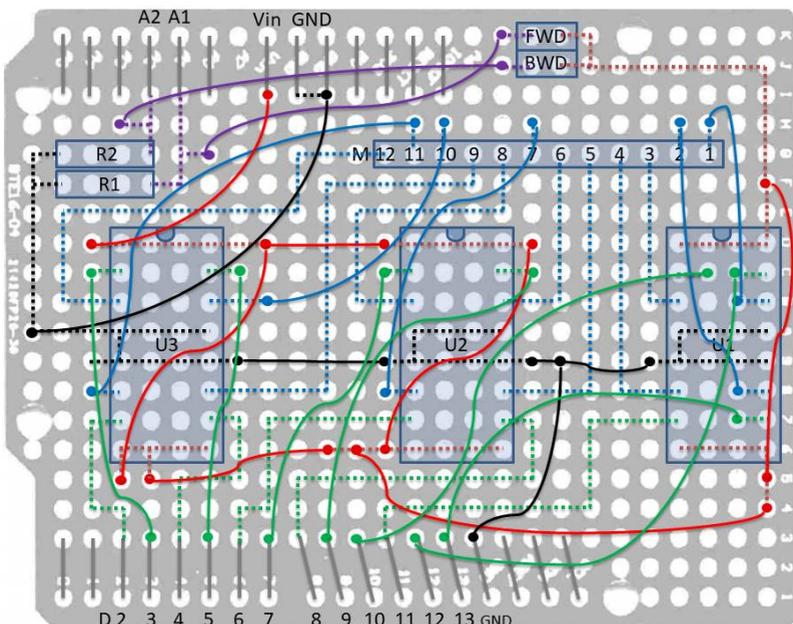


Figura 3.52 – Esquema da placa ilhada vista de cima. Fonte: Dos autores.

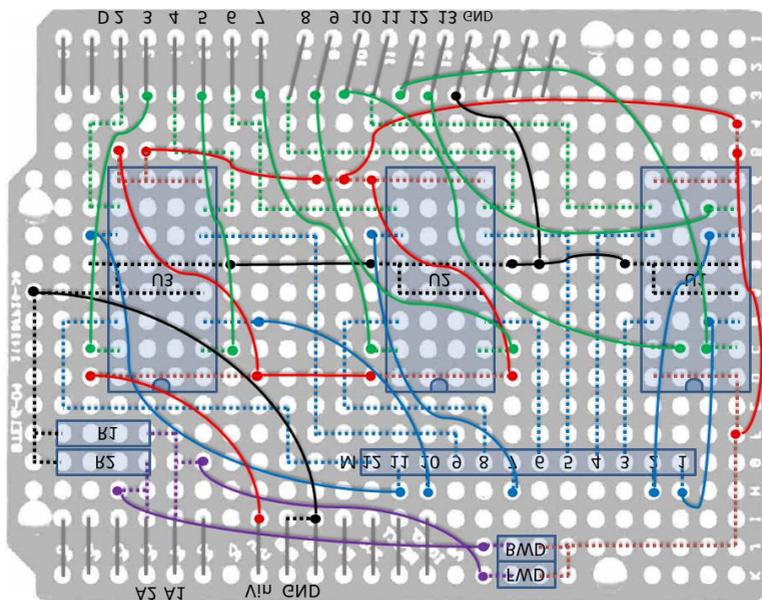


Figura 3.53 – Esquema da placa ilhada vista de baixo (espelhada). Fonte: Dos autores.

Sugere-se o seguinte roteiro para a soldagem:

1. Pelo lado de baixo, soldar apenas os **pontos das ilhas de continuação**. Ou seja, soldar apenas as ilhas referentes à continuação de trilhas que não correspondam a terminais de componentes e nem a entradas de fios. Inicie com as de cor preta, depois as vermelhas, depois as verdes, depois as azuis. A cada passo confira se está ficando correto. Vide a Figura 3.54.

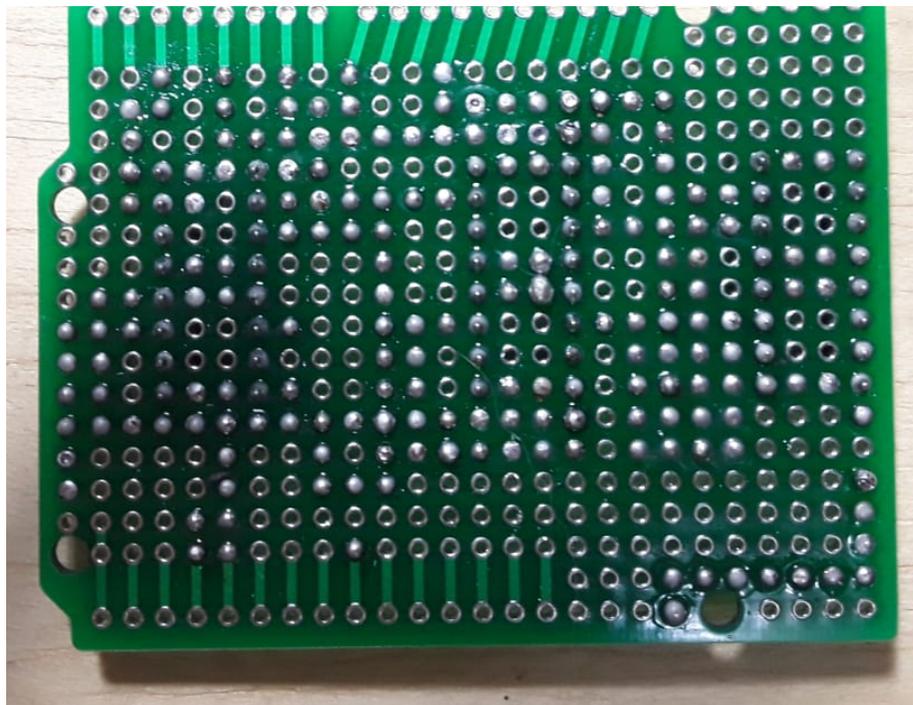


Figura 3.54 – Soldar as ilhas de continuação.

Durante a soldagem dos pontos, deve-se seguir o *Esquema da placa ilhada vista de baixo* indicado anteriormente e impresso em uma folha de papel. A cada trecho soldado, deve-se ir dando um visto à caneta na folha impressa conforme ilustra a Figura 3.55.

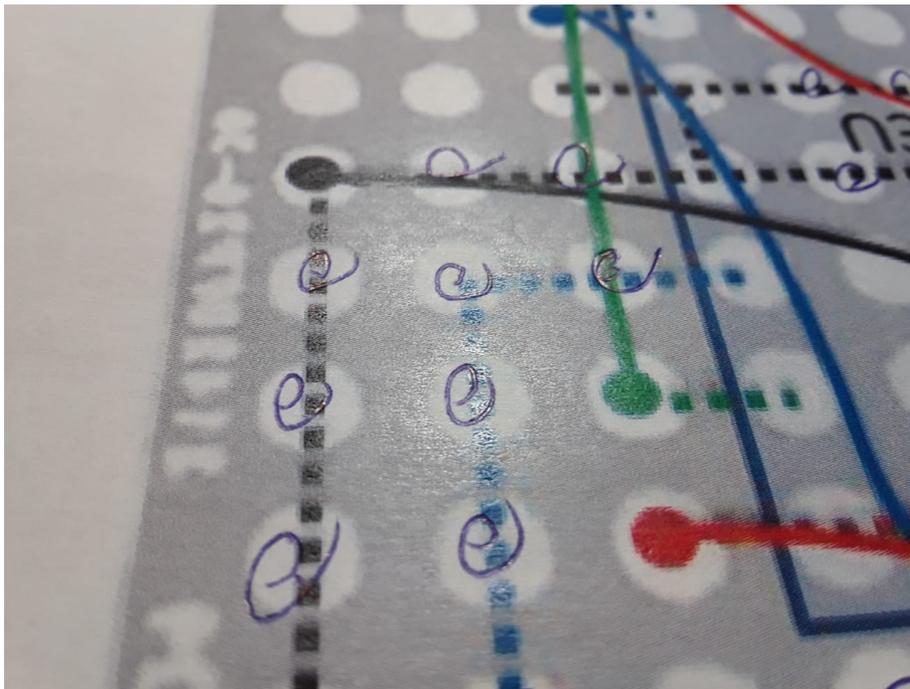


Figura 3.55 – Visto em trechos do esquema impresso.

2. Pelo lado de baixo, solde os **terminais dos soquetes dos CIs**. Ou seja, a soldagem é por baixo, mas os soquetes estão por cima da placa. Lembre-se de deixar o soquete na direção correta, com o entalhe em U voltado adequadamente, conforme as figuras dos esquemas. A Figura 3.56 apresenta o resultado desta etapa visto de baixo e a Figura 3.57, visto de cima.

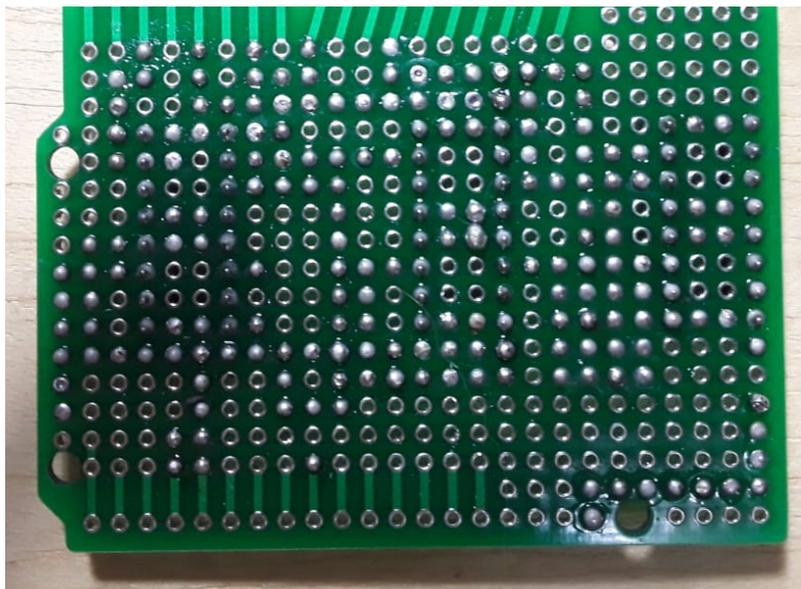


Figura 3.56 – Pontos dos terminais dos soquetes soldados.

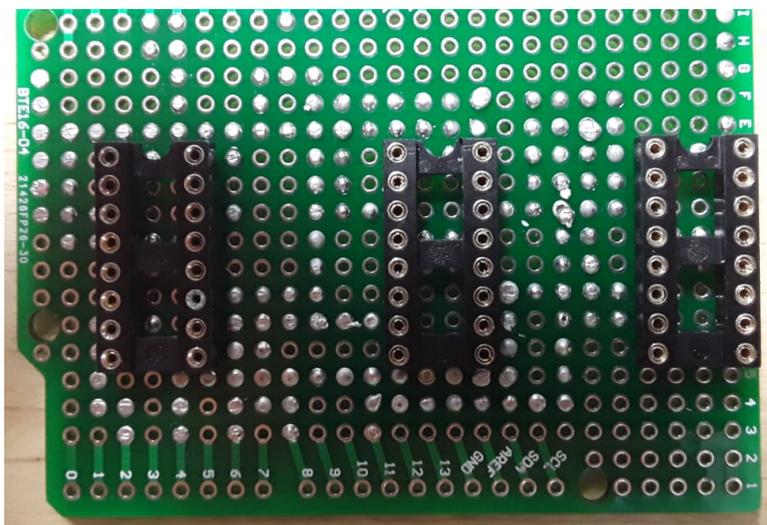


Figura 3.57 – Terminais dos soquetes soldados.

3. Pelo lado de baixo, solde os **terminais dos resistores**. Ou seja, a soldagem é por baixo, mas os resistores estão por cima da placa. Após a soldagem, corte o excesso dos terminais dos resistores. Vide a Figura 3.58.

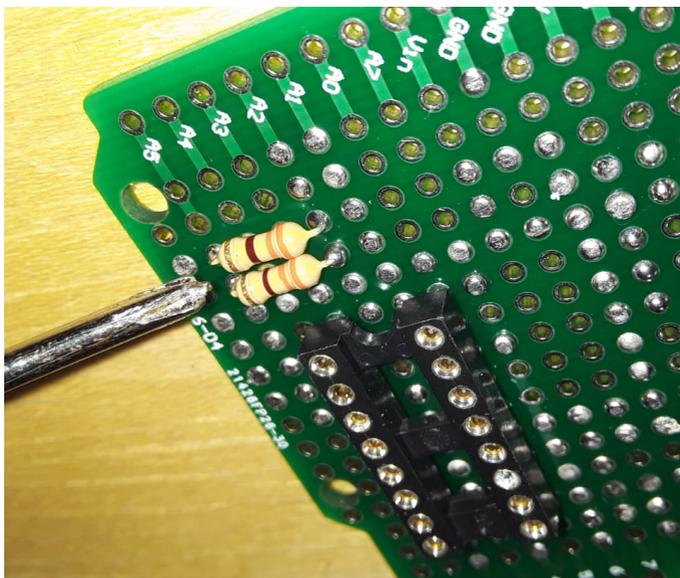


Figura 3.58 – Resistores soldados.

4. Pelo lado de baixo, solde os **terminais do conector dos motores**. Ou seja, a soldagem é por baixo, mas os terminais estão por cima da placa. Vide a Figura 3.59.



Figura 3.59 – Conectores para os motores soldado.

5. Pelo lado de baixo, solde os **terminais dos conectores dos botões**. Ou seja, a soldagem é por baixo, mas os terminais estão por cima da placa. Vide a Figura 3.60.

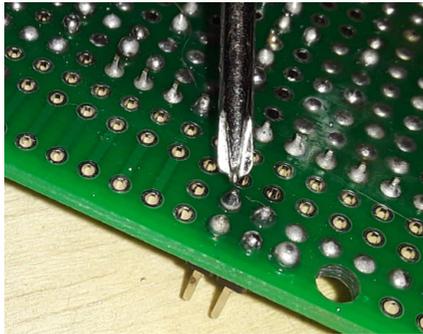


Figura 3.60 – Conectores para os botões soldado.

6. Pelo lado de cima, solde os **terminais dos conectores que vão encaixar sobre o Arduino**. Ou seja, a soldagem é por cima, mas os terminais estão voltados para baixo da placailhada. Uma dica é encaixar a placailhada no Arduino antes da soldagem para garantir que o encaixe seja perfeito. Veja a Figura 3.61 que apresenta o Arduino com os terminais encaixados para guiar a soldagem. Depois de soldar, retire novamente a placa de cima do Arduino. O Resultado da soldagem é apresentado nas Figuras 3.62 e 3.63.



Figura 3.61 – Arduino disposto como guia para a soldagem.

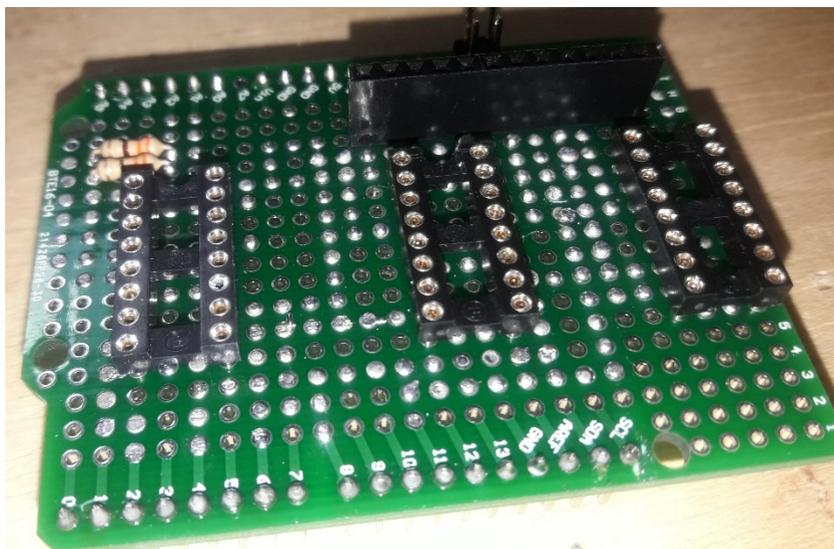


Figura 3.62 – Terminais dos conectores que vão encaixar sobre o Arduino soldados, visto de cima.

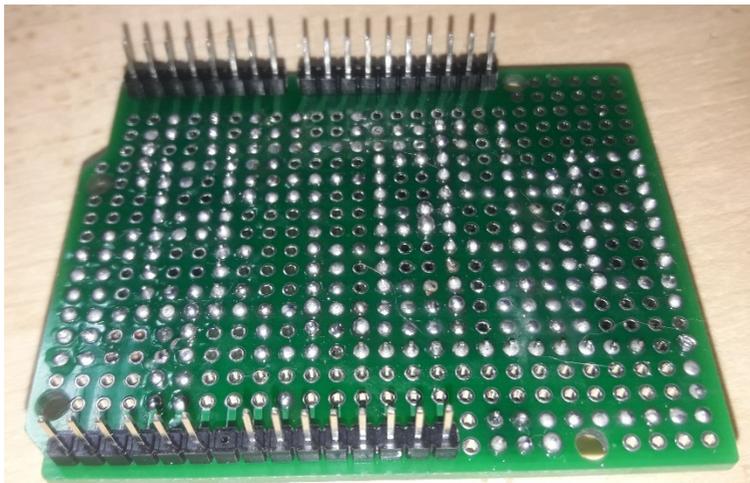


Figura 3.63 – Terminais dos conectores soldados, que vão encaixar sobre o Arduino, visto de baixo.

7. Pelo lado de baixo, complete as trilhas soldando os **espaços entre as ilhas** previamente soldadas. Ainda não solde onde entrará fios. Vide a Figura 3.64 para observar o resultado da soldagem. Este é o processo mais difícil de todos nesta placa ilhada e, por isso, requer paciência e destreza do soldador.

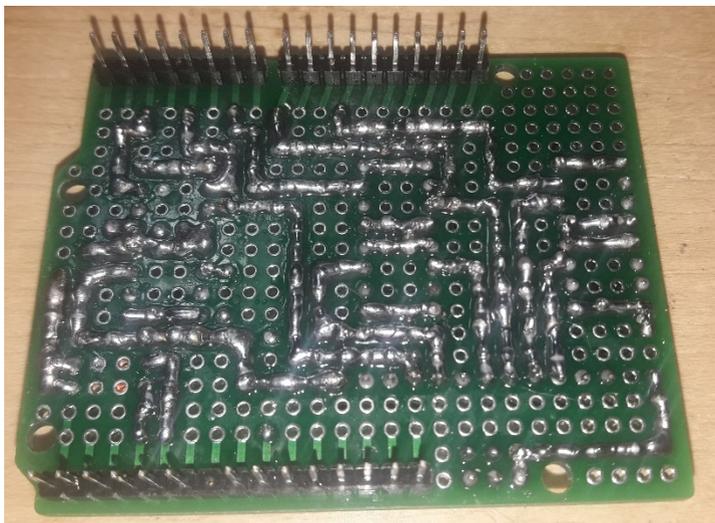


Figura 3.64 – Espaços entre as ilhas soldados.

Também é muito importante as anotações de cada soldagem no Esquema impresso em papel. Conforme ilustra os Xs na Figura 3.65.



Figura 3.65 – Vistos em X para os espaços entre as ilhas soldados.

8. Insira cada **fio preto** pelo lado de baixo e o solde pelo lado de cima. Busque deixar os fios curtos para não entulhar muito o espaço. Após colocar cada ponta de fio, já complete a trilha soldando os espaços entre as ilhas e a pontas recém soldadas. Vide a Figura 3.66.

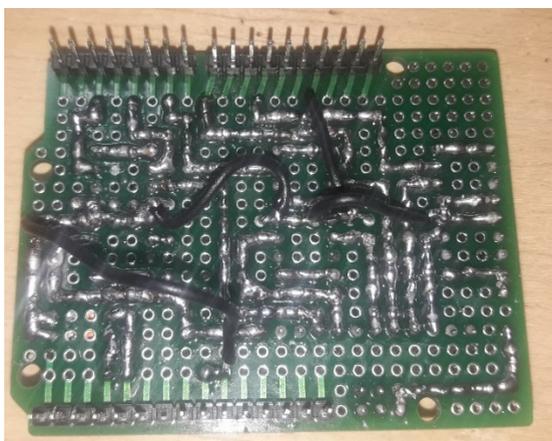


Figura 3.66 – Fios pretos soldados.

9. Insira cada fio vermelho pelo lado de baixo e o solde pelo lado de cima. Após colocar cada ponta de fio, já complete a trilha soldando os espaços entre as ilhas e a pontas recém soldadas. Vide a Figura 3.67.

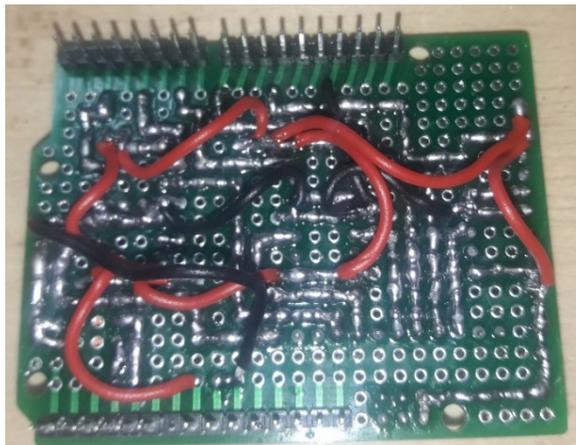


Figura 3. 67 – Fios vermelhos soldados.

10. Insira cada fio verde pelo lado de cima e o solde pelo lado de baixo. Após colocar cada ponta de fio, já complete a trilha soldando os espaços entre as ilhas e a pontas recém soldadas. Vide a Figura 3.68.

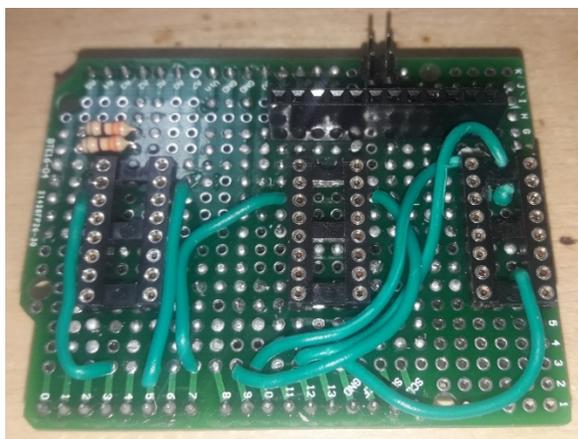


Figura 3. 68 – Fios verdes soldados.

11. Insira cada fio azul pelo lado de cima e o solde pelo lado de baixo. Após colocar cada ponta de fio, já complete a trilha soldando os espaços entre as ilhas e a pontas recém soldadas. Vide a Figura 3.69.

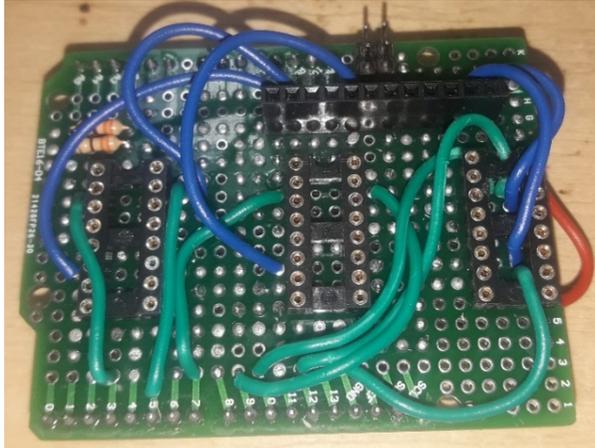


Figura 3. 69 – Fios azuis soldados.

12. Insira cada fio roxo pelo lado de cima e o solde pelo lado de baixo. Após colocar cada ponta de fio, já complete a trilha soldando os espaços entre as ilhas e a pontas recém soldadas. Vide a Figura 3.70.

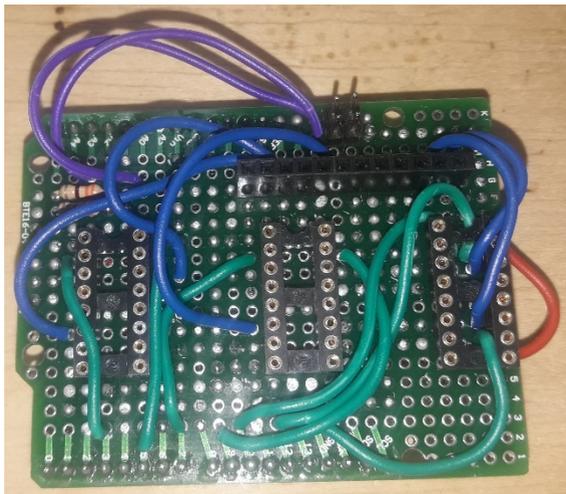


Figura 3. 70 – Fios roxos soldados.

13. Confira se está tudo certo, conforme o esquema.

Após terminado o trabalho, confira se não existe algum curto no circuito por meio de um multímetro, para verificar a resistência. Indica-se os seguintes pontos de teste.

1. No GND da placa coloque um dos terminais do multímetro e o outro vá verificando se a Resistencia é 0 ohms entre todos os pontos e fios pretos. Depois, teste em outros locais, pois deverá dar infinito.
2. No Vin da placa coloque um dos terminais do multímetro e o outro vá verificando se a Resistencia é 0 ohms entre todos os pontos e fios vermelhos. Depois, teste em outros locais, pois deverá dar infinito.
3. Teste cada trilha e fio correspondente às portas digitais do Arduino (de D2 a D13) e também às portas A1 e A2.
4. Teste cada trilha e fio correspondente aos motores.

Finalmente, após conferir tudo e ter certeza de que não há nenhum problema, encaixe os CIs no lugar certo, com o entalhe em U voltado para o lado correto. Por fim, encaixe a placa sobre o Arduino e ligue os terminais dos botões nas suas conexões.

Para a alimentação, utilize uma fonte DC de **5V** na Entrada de alimentação do Arduino, não utilize mais que 5V pois irá queimar os motores da célula. Como não está sendo utilizada a saída de 5V do Arduino (5V) e sim o **Vin** para alimentar os motores, isso significa que não existe nenhuma regulagem de voltagem. Ou seja, o Arduino não vai corrigir um erro de voltagem a mais nos motores e eles iram queimar. Por isso, utilize apenas uma fonte DC de 5V na Entrada, com o positivo no centro e o negativo nas extremidades ().

A Figura 3.71 mostra o resultado final da placailhada soldada por cima e a Figura 3.72 a apresenta por baixo.

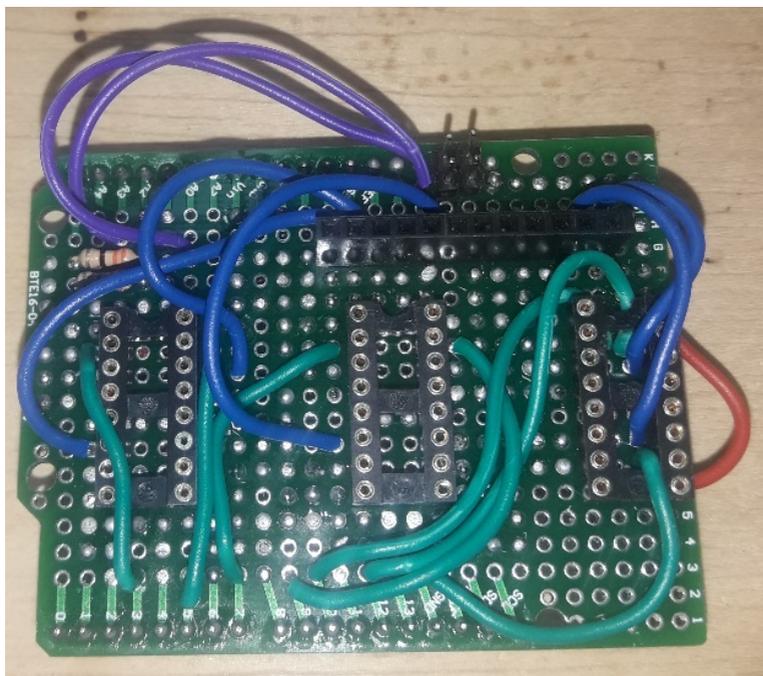


Figura 3.71 – Resultado final da placailhada por cima. Fonte: Dos autores.

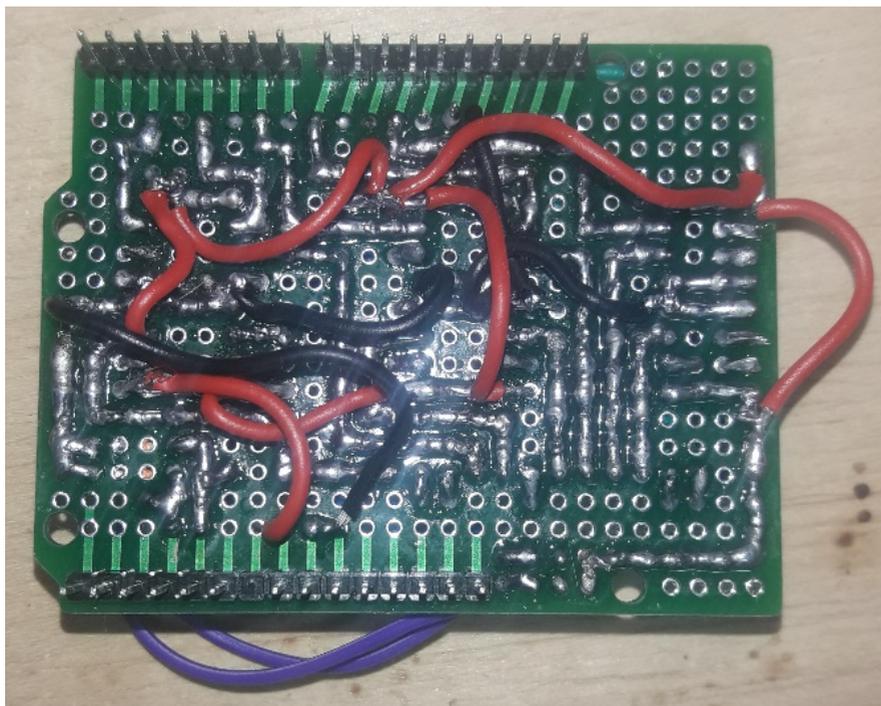


Figura 3.72 – Resultado final da placailhada por baixo. Fonte: Dos autores.

Antes de ligar a placailhada no Arduino alimentado pela força (Fonte de alimentação ou cabo UAB), é importante **desinstalar** qualquer aplicação previamente instalada no Arduino que não seja a aplicação correta do Celta, conforme veremos no Capítulo 4. Para “desinstalar” um programa (sketch) do Arduino, na verdade você deve carregar um sketch vazio na placa. Basta abrir a IDE do Arduino e criar um Novo sketch, conectar o Arduino, selecionar a porta e mandar carregar este sketch vazio. Se você não tomar esta precaução, pode danificar sua placa, seus motores e seu Arduino.

A Figura 3.73 apresenta a placailhada já encaixada sobre o Arduino.

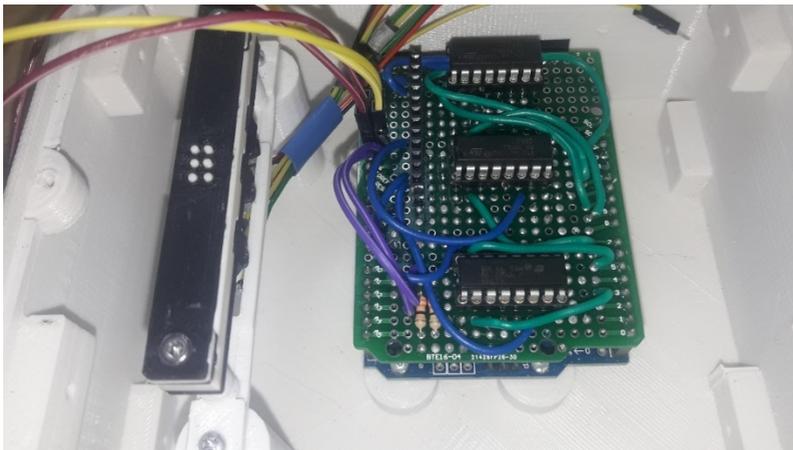


Figura 3.73 – Placailhada finalizada sobre o Arduino. Fonte: Dos autores.

3.4.4 Confeccionando uma placa específica para o Celta

Se você desejar pode optar pela confecção de uma placa de circuito impresso específica para o Celta. Nesta seção, iremos explicar um pouco sobre esse processo.

Para a confecção da placa, o esquema elétrico simplificado apresentado na Figura 3.49 foi transformado, por meio do software EasyEDA, em um desenho de placa, conforme ilustra a Figura 3.74.

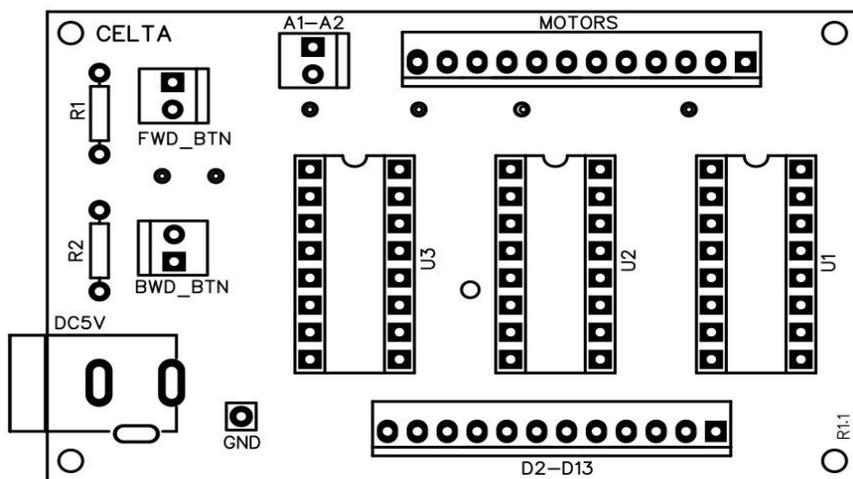
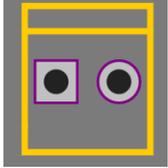
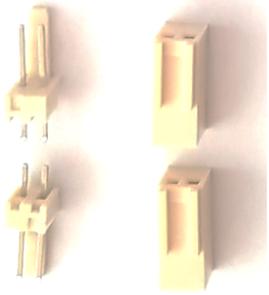
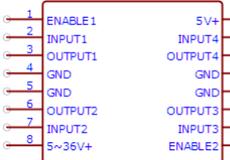
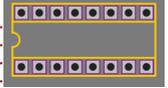
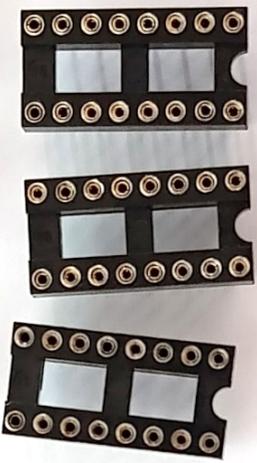


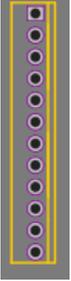
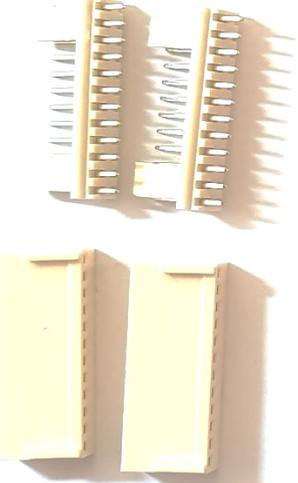
Figura 3.74 – Visão frontal da PCI com as legendas.

O desenho elaborado no EasyEDA é de uma placa de dupla face impressa em uma impressora laser em transparência e posteriormente aderida à placa cobreada em ambas as faces por meio de uma luz UV e, por fim, após um processo químico de corrosão, restará apenas as trilhas cobreadas na placa.

O Quadro 3.7 apresenta a descrição e a simbologia dos componentes da Figura 3.49, a simbologia na imagem de placa produzida no EadyEDA (Figura 3.75) e a fotografia do componente.

Quadro 3.7 – Simbologia e componentes para PCI.

Descrição e Legenda na Placa	Simbologia no diagrama	Simbologia na imagem da placa	Fotografia
DIP-1X2P-2.54MM (A1-A2, FWD_BTN e BWD_BTN)			
Soquete DIP-16 (U1, U2 e U3)			

Resistor de 330 ohms, 5%, 1/4W (R1 e R2)			
Conector DIF 2510S-12P, para encaixar no Arduino. (MOTORS e D2-D13)			
Diversos cabos flexíveis com conectores Dupont macho e fêmea			
Placa folheada de cobre, dupla face para PCB			

Como são muitas ligações e conexões entre os circuitos integrados e as saídas e as entradas, optamos por confeccionar uma “placa de circuito impresso de dupla face”. Além disso,

achamos por bem utilizar o “método de máscara fotossensível com UV”. Para entender mais sobre esses elementos, busque por videoaulas no *Youtube*, são bastante esclarecedoras.

A Figura 3.75 apresenta a imagem da placa gerada para a impressão do fotolito, que será utilizado para marcar os locais onde não ocorrerá a corrosão do cobre. Ou seja, onde serão as trilhas.

Observamos que sobre o fundo preto quadriculado da tela do software temos as trilhas em vermelho, relativas a um dos lados da placa, e as trilhas em azul, relativas à outra face da placa. Na figura, podemos ver ainda algumas marcações em cor laranja, estas são apenas referências à localização dos componentes para as ligações, ou seja, não fazem parte das trilhas de condução elétrica.

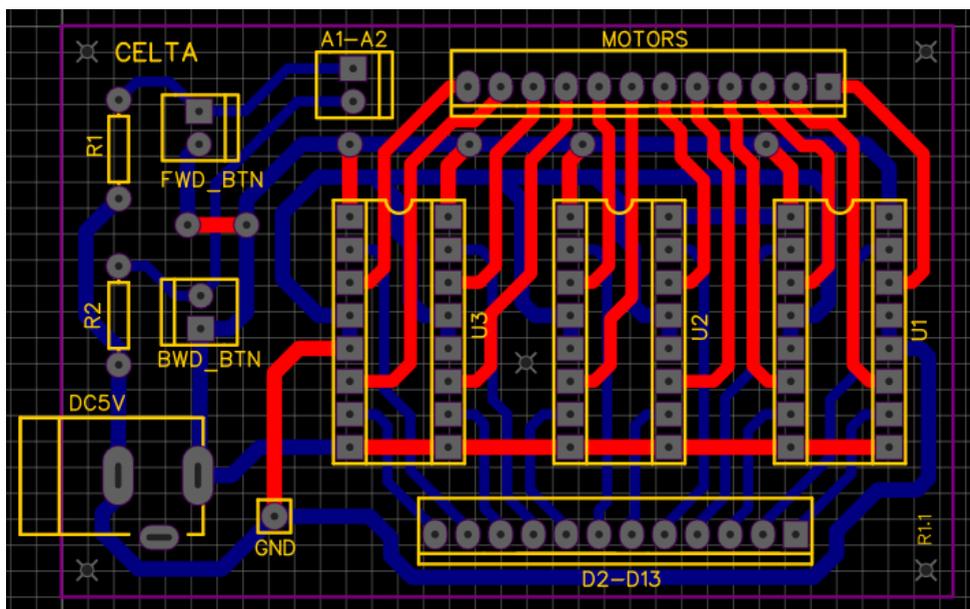


Figura 3.75 – Visão do projeto da placa no software.

A Figura 3.76 apresenta o que será impresso no lado da frente na máscara para a corrosão.

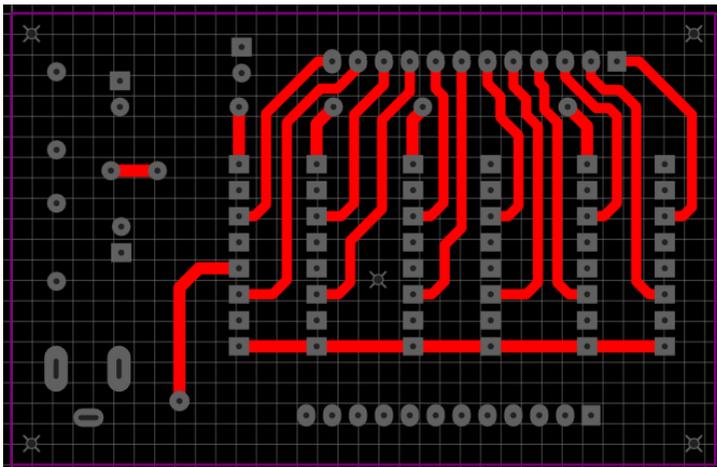


Figura 3.76 – Visão frontal do conteúdo para máscara.

A Figura 3.77, por sua vez, apresenta a visão traseira.

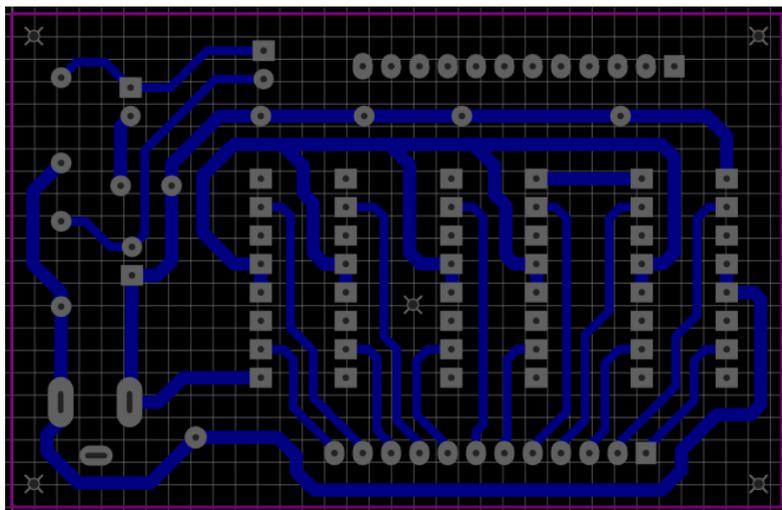


Figura 3.77 – Visão frontal do conteúdo para máscara.

É importante lembrar que essa placa já está no tamanho específico, isto é, caso não siga as medidas estabelecidas, não será possível encaixar no Arduino e nos conectores.

Disponibilizamos um arquivo PSD (que é o photoshop) e também uma mesma versão em PDF e outros formatos, no seguinte link:

[<https://celta.ufscar.br/repositorio-de-arquivos/>](https://celta.ufscar.br/repositorio-de-arquivos/)

Basta imprimir o arquivo com uma impressora laser, em preto, uma transparência. Nesse arquivo, temos 4 imagens na esquerda e mais 4 imagens na direita. Elas são ligeiramente diferentes entre a esquerda e a direita. De cima para baixo, as imagens estão duplicadas. A ideia é você imprimir a folha toda e recortar cada um dos 8 desenhos. Depois utilizar uma máscara dupla com os desenhos em preto, um tipo de cada lado. Vide a Figura 3.78 para entender melhor esta explicação.

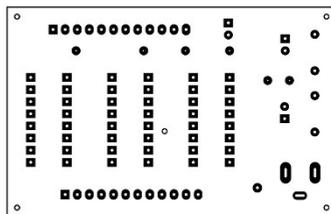
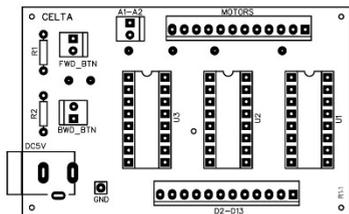
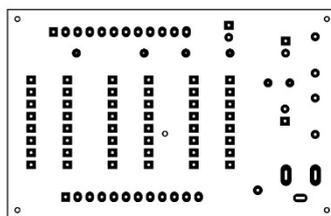
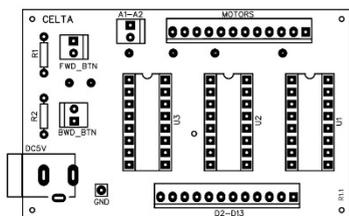
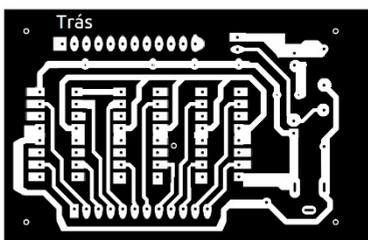
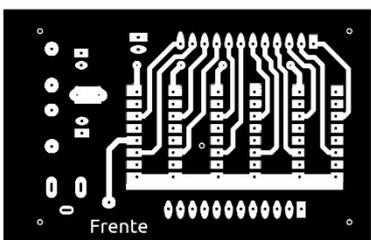
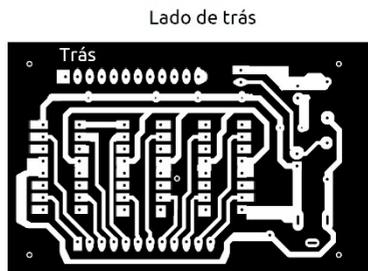
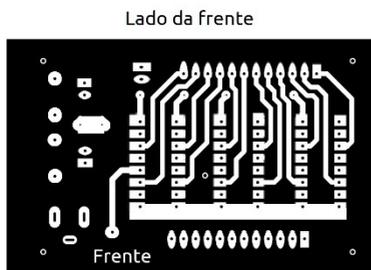


Figura 3.78 – Imagens para impressão em transparência para a máscara.

A Figura 3.79 apresenta uma fotografia com o resultado final da impressão da transparência, já com os 8 desenhos recortados.



Figura 3.79 – Desenhos para as máscaras imprimidos em transparência.

A ideia de colocar camadas duplas na máscara é apenas para garantir que a luz ultravioleta não ultrapasse a máscara. Às vezes, com uma única camada, ocorre algumas pequenas falhas, então por isso indicamos a utilização destas duas camadas em cada lado da placa. Vide a Figura 3.80 para observar a aplicação da luz ultravioleta na máscara, na placa.

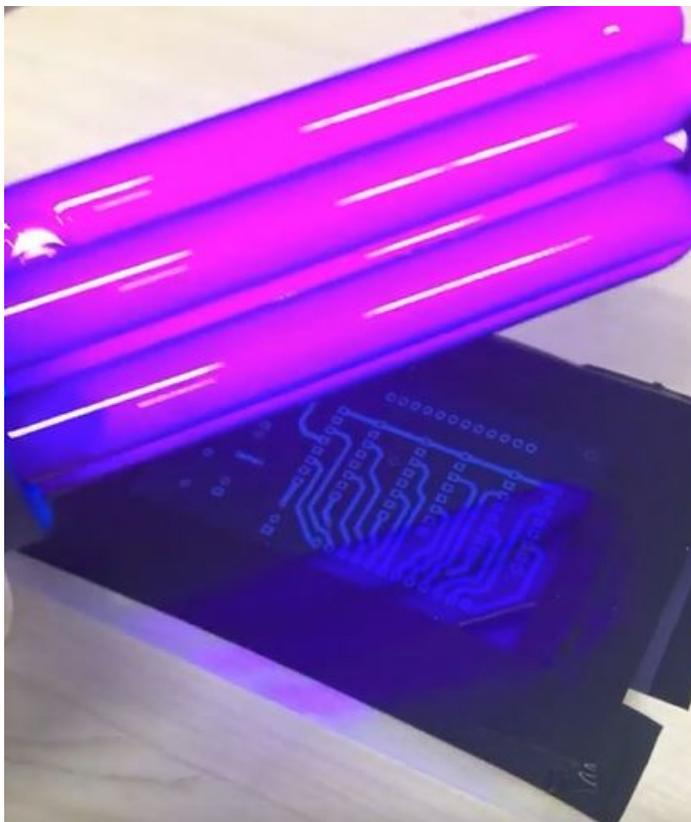


Figura 3.80 – Aplicação de luz ultravioleta na face frontal da placa.

Após a aplicação da máscara com a luz ultravioleta, a placa fica pronta para ser corroída, com as partes de contato elétrico protegidas, conforme se observa na Figura 3.81.

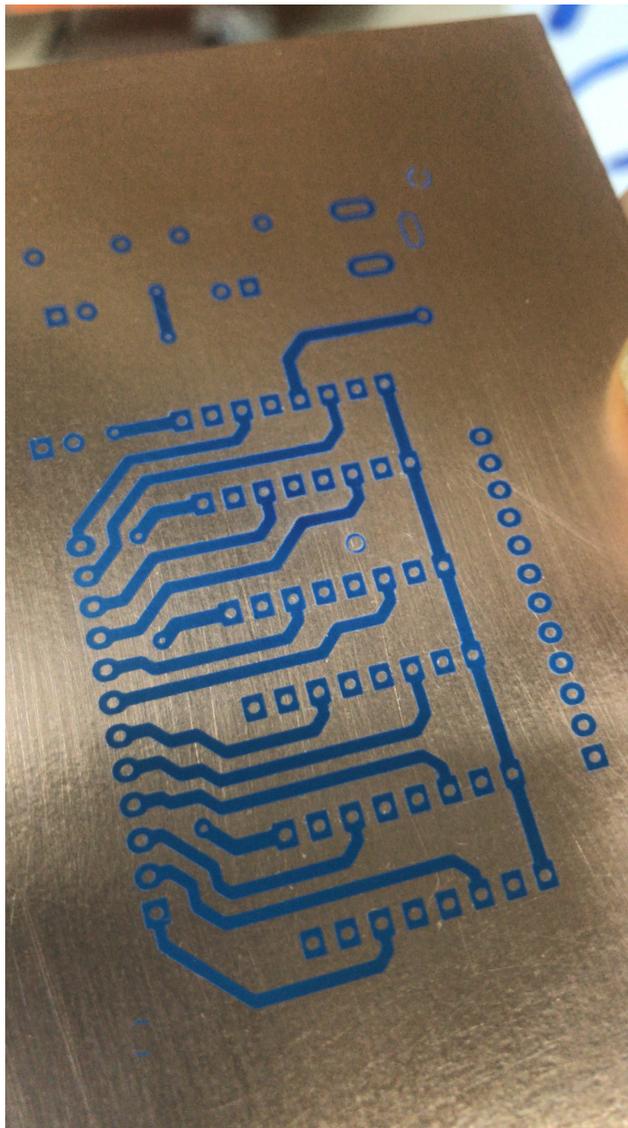


Figura 3.81 – Face frontal da placa de cobre pronta para ser corroída.

A Figura 3.82 apresenta uma das faces da placa pronta, após o processo de corrosão.

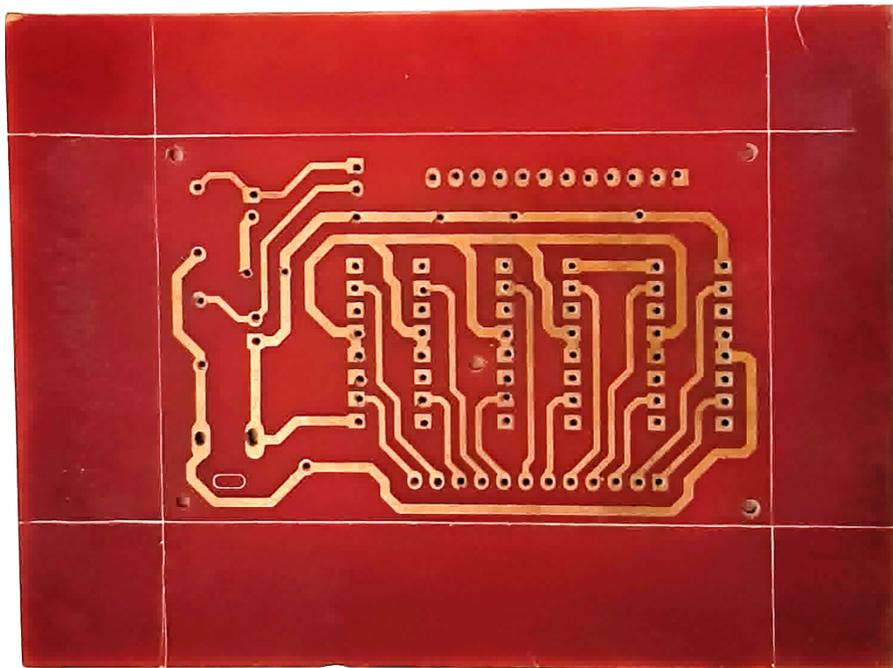


Figura 3.82 – Face traseira da placa de cobre após corrosão.

Mais anteriormente, na Figura 3.78, tínhamos ainda os quatro desenhos em branco, abaixo. A ideia é depois que a placa estiver corroída, você fazer em silkscreen a marcação destas imagens em cada lado da placa. Isso proporciona um acabamento melhor em sua placa e facilita depois a montagem dos componentes, mas isso é dispensável. Veja na Figura 3.83 como fica a impressão desta máscara para o silkscreen.

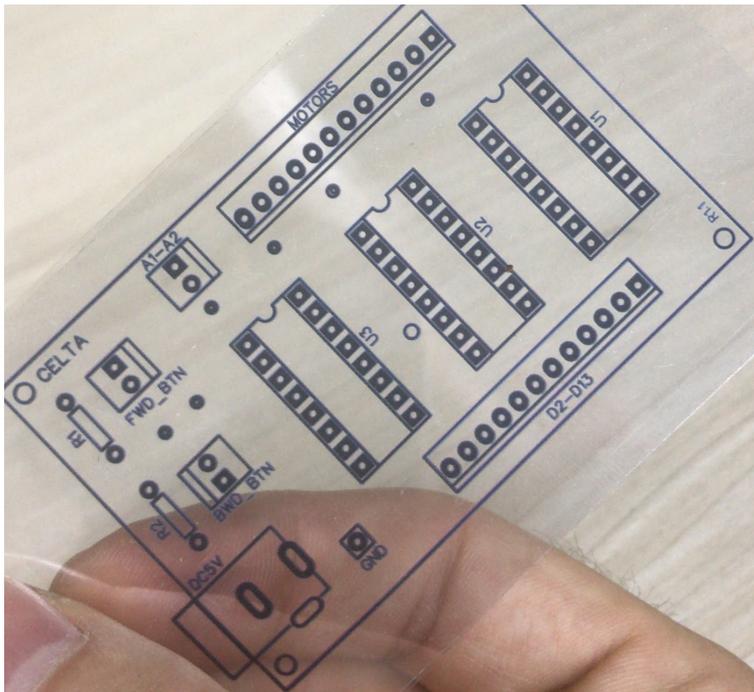


Figura 3.83 – Transparência impressa com a legenda das conexões e componentes.

Para entender melhor sobre este tópico, até o que foi explanado no momento, assista ao vídeo disponibilizado no seguinte link:

<https://www.youtube.com/watch?v=QGipBsTa0vI>.

Montagem dos componentes na placa demanda cuidados pois é necessário de seja garantido o contato elétrico em ambas as faces da placa. A recomendação é que, antes da soldagem dos componentes, se faça uma conexão de solda entre uma camada e outra da placa e com um chupador de solda se retire esta soldagem. Assim, o orifício fica com uma conexão entre os lados. Para isso dar certo, pode ser necessário alargar um pouco os orifícios por meio de uma broca bem fina.

Em nosso protótipo deixamos para fazer o corte da placa após a montagem de alguns terminais. Isso não é recomendado pois o

ideal seria cortar a placa antes da soldagem dos terminais. De qualquer forma, a Figura 3.84 ilustra o resultado final do processo de corte da placa, na qual utilizou-se uma serra de arco manual e um torno de bancada para auxiliar no processo.



Figura 3.84 – Bancada ao se cortar a placa.

Se você comparar a Figura 3.78, Imagens para impressão em transparência para a máscara, com a fotografia nas Figuras 3.85 e 3.86 verá ligeiras mudanças. É que aprimoramos o modelo de máscara para facilitar o seu futuro trabalho de soldagem.

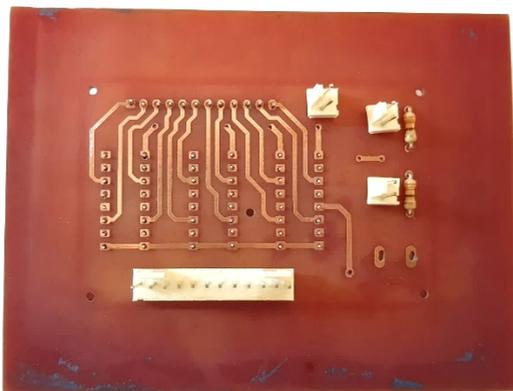


Figura 3.85 – Face frontal da placa com terminais encaixados, conforme projeto inicial.

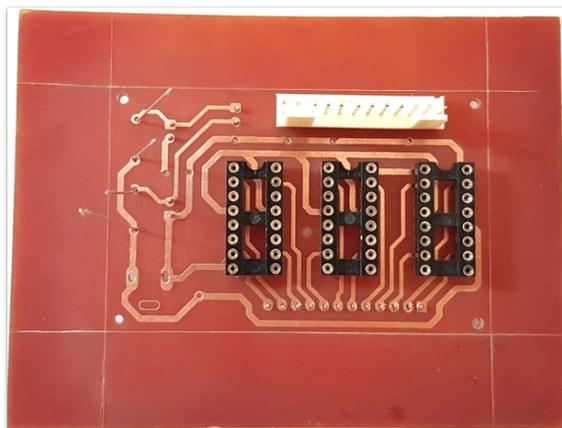


Figura 3.86 – Face traseira da placa com terminais encaixados conforme projeto inicial.

Em nosso protótipo tivemos dificuldade na soldagem, pois as trilhas estavam muito pequenas. O que fizemos, na prática, foi inverter a posição do CI central (o U2)¹⁹. Ele ficou encaixado na parte frontal e não na parte traseira, como ilustra as Figuras 3.85 e 3.86. Ou seja, não aconselhamos que se siga estas figuras e sim o resultado final que obtivemos, conforme as Figuras 3.87 e 3.88.

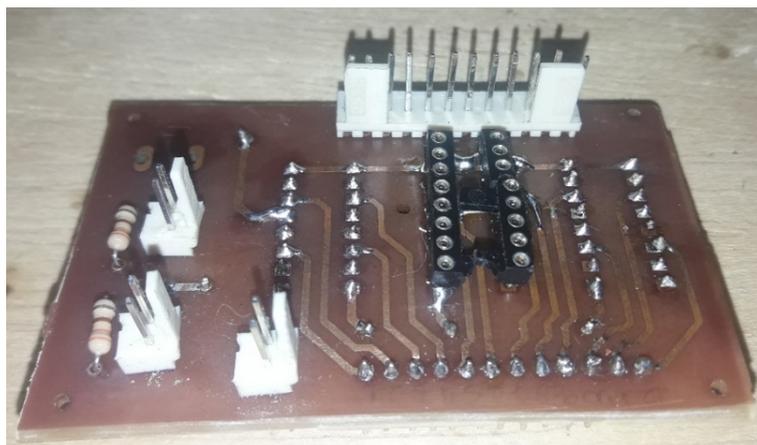


Figura 3.87 – Face frontal da placa finalizada.

¹⁹ Observe apenas que ao fazer isso os motores 3 e 4 ficam trocados. Mas isso não é problema, basta trocar os fios dos motores nos conectores durante o ajuste fino após a montagem de tudo.

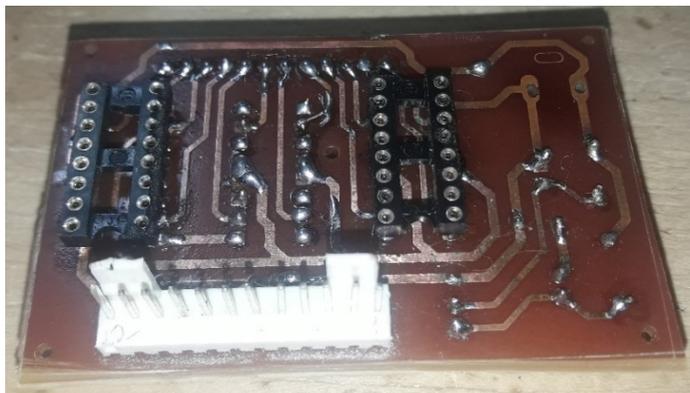


Figura 3.88 – Face traseira da placa finalizada.

Resumindo a soldagem: Solde os resistores, depois os soquetes do CI 1, depois do CI2 e por fim do CI3. E, finalmentente, solde os conectores. A placa ficará conforme ilustrado nas Figuras 3.87 e 3.88.

3.5 Preparação dos Motores

Existem diversos tipos de motores para vibração de celulares. Para o uso sem nenhuma adaptação no Celta, é necessário que se adquira, exatamente, motores com a seguinte especificação. DC1.5-3V, com 4mm de diâmetro, com 8mm de comprimento, com roda excêntrica de 4.4x4.4mm e voltagem DC de 1.5 a 3V (busque na Internet pelo seguinte produto: *DC Micro Vibration Motor Cell Phone Coreless Vibrator 4.4x4.4mm*). A Figura 3.89 apresenta uma fotografia de um destes motores.



Figura 3.89 – Motor a ser utilizado no projeto.

Observe que, embora a alimentação indicada do motor seja de até 3V, em nosso circuito utilizamos 5V. Com essa alimentação a mais, o motor consegue um melhor desempenho no Celta e, como ele fica submetido a esta tensão adicional por um período de tempo muito curto, ele não queima.

A primeira coisa a fazer para preparar os motores é organizar os materiais e o espaço de trabalho. Como são componentes muito pequenos, será difícil o manuseio sem estes cuidados e uma iluminação adequada. A Figura 3.90 mostra o local de preparação dos motores que utilizamos no projeto. Vê-se os elementos necessários: Retífica, mini morsa, ferro de soldar, estanho, paquímetro digital etc.

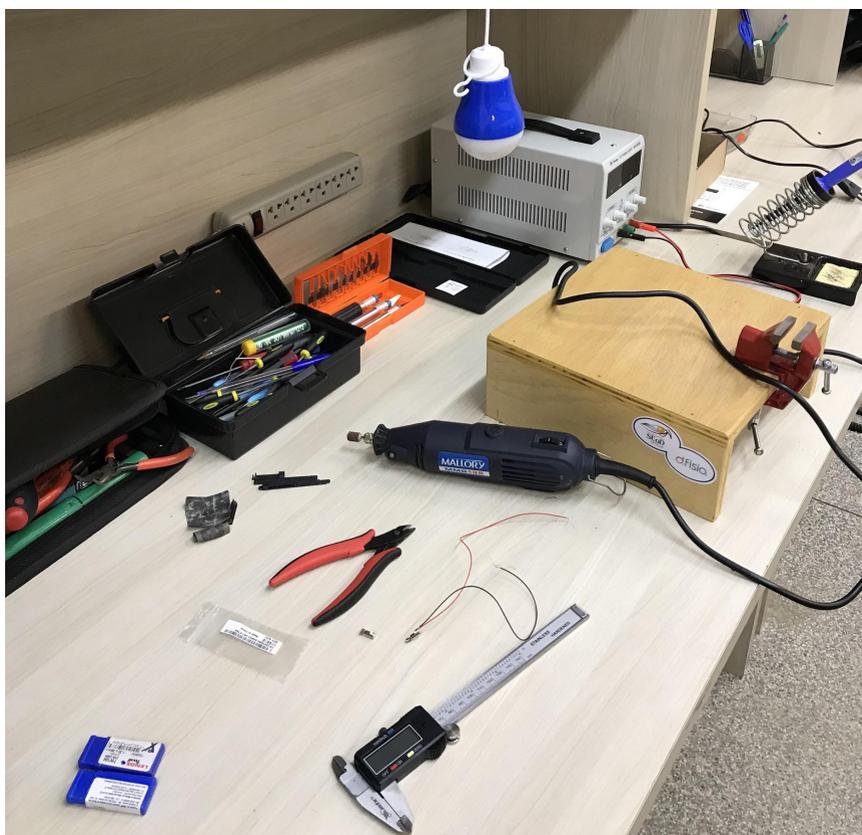


Figura 3.90 – Visualização da bancada de trabalho para preparação dos motores.

O motor é provido de uma cabeça excêntrica. Ou seja, é uma cabeça com meio cilindro de formação. Vide a Figura 3.91 para entender como é a cabeça original do motor.



Figura 3.91 – Detalhe da cabeça giratória de um motor com as bordas originais.

Então, a primeira tarefa a ser realizada é desbastar uma das pontas da cabeça giratória, como nas Figuras 3.92 e 3.93. Com muito cuidado, prenda parte da cabeça do motor em uma mini morsa e com uma lima ou uma ponta de esmeril em uma mini retífica, deixe uma das bordas desbastada.



Figura 3.92 – Detalhe da cabeça giratória de um motor com uma das bordas desbastadas com a retífica (Dremel).



Figura 3.93 – Detalhe da cabeça giratória de um motor com uma das bordas desbastadas.

Realize esta operação em todos os 6 motores. A Figura 3.94 mostra diversos motores, já com um dos lados das cabeças desbastadas.



Figura 3.94 – Vários motores já com um dos lados da cabeça desbastados.

O próximo passo é preparar os cabos de conexão. Usa-se cabos flexíveis (24 AWG, 0,20mm) em cores variadas. Na verdade, não será necessário você se preocupar com a polaridade destes motores, pois o ajuste fino de qual será o pino correto a ser ligado na placa de controle para cada motor só será estabelecido após os primeiros testes com o Celta já todo montado. Preocupe-se apenas em distinguir um motor do outro. Pense em usar cores diferentes para cada motor. Separe 12 pedaços de cerca de 15 cm destes cabos. E encaixe e solde em uma das pontas de cada pedaço um conector Dupont fêmea, como ilustra a Figura 3.95.

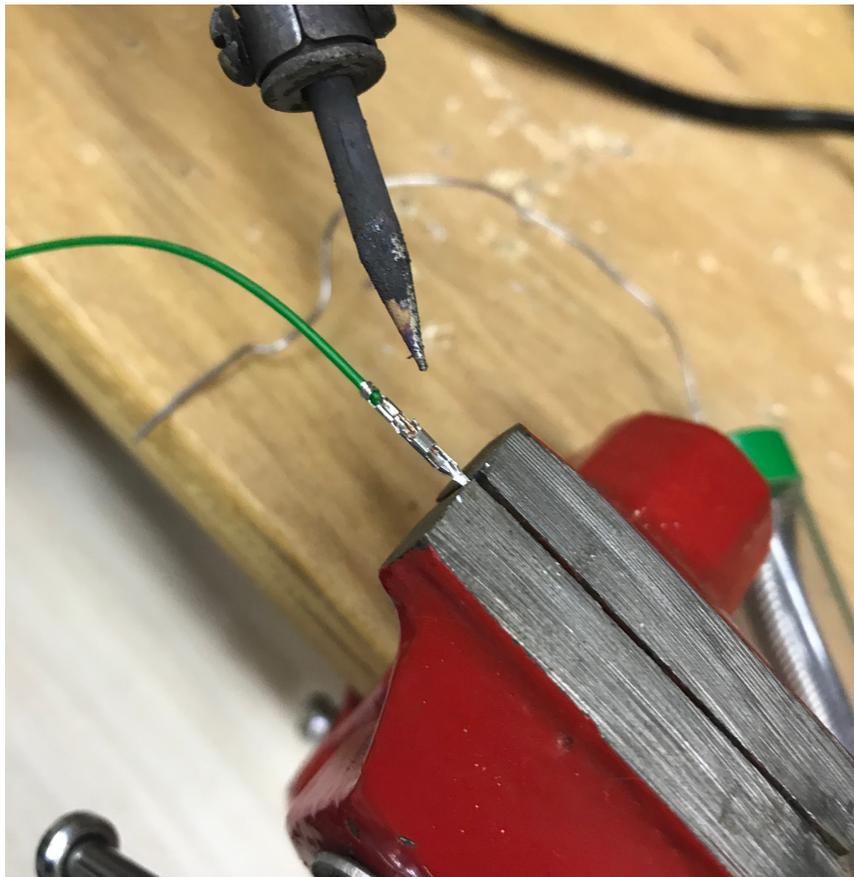


Figura 3.95 – Soldando um dos cabos dos motores ao conector.

Depois de soldar as pontas, basta encaixar o invólucro protetor preto do conector Dupont que pode ser visualizado, ainda fora do lugar, na Figura 3.96.²⁰

²⁰ Observe que no lugar de fazer esta montagem inicial do cabo você pode simplesmente utilizar um cabo Dupont Macho-Macho e cortar um dos lados.

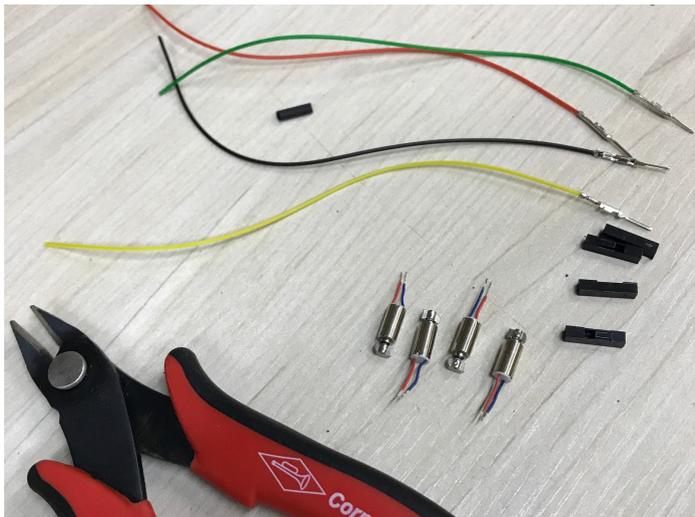


Figura 3.96 – Preparando para conexão dos fios aos motores.

Depois, com muito cuidado, solde a outra extremidade dos fios a cada terminal dos motores. Utilize um isolante termo retrátil para fazer o isolamento. Ficará conforme ilustra a Figura 3.97.

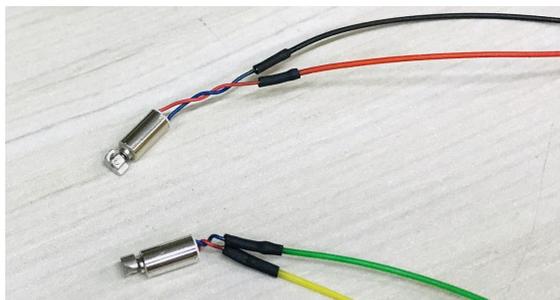


Figura 3.97 – Vários motores com os cabos soldados e isolados.

Antes da soldagem dos cabos nos motores, indicamos que você proteja os fios dos motores com uma proteção que elaboramos na modelagem e impressão em 3D, chamamos esta proteção de *cases*. A Figura 3.98 apresenta um motor com um desses *cases* encaixado na parte de baixo. No dia a dia de nosso projeto verificamos que esta proteção é útil para evitar a quebra dos conectores dos motores, que são muito frágeis.

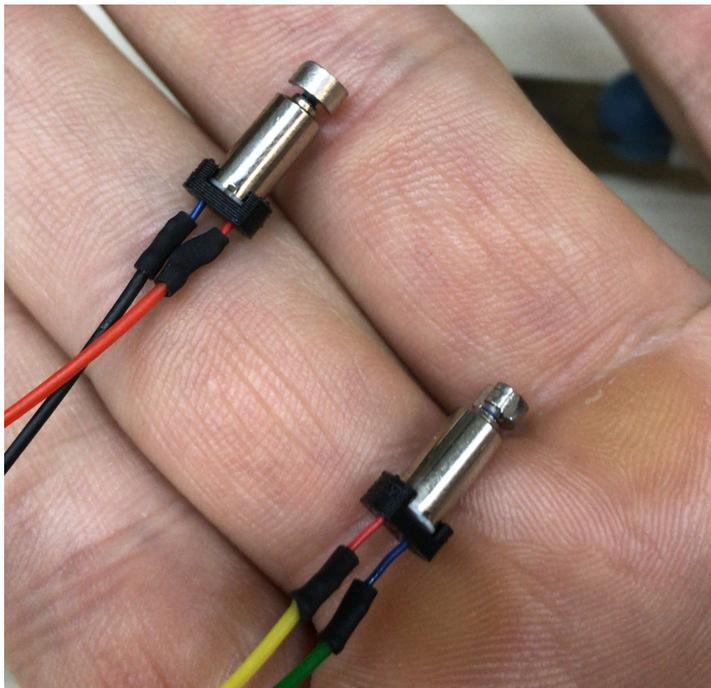


Figura 3.98 – Motores já com os cabos e com cases de proteção.

3.6 Montagem final

Com os motores prontos, a próxima fase é montagem da célula. Para isso, pegue todas as peças impressas na impressora 3D e mais dois parafusos de 30mmx2mm, com porcas. Conforme ilustra a Figura 3.99.



Figura 3.99 – Parafusos compridos com porca.

Para os próximos passos, considere a Figura 3.100.

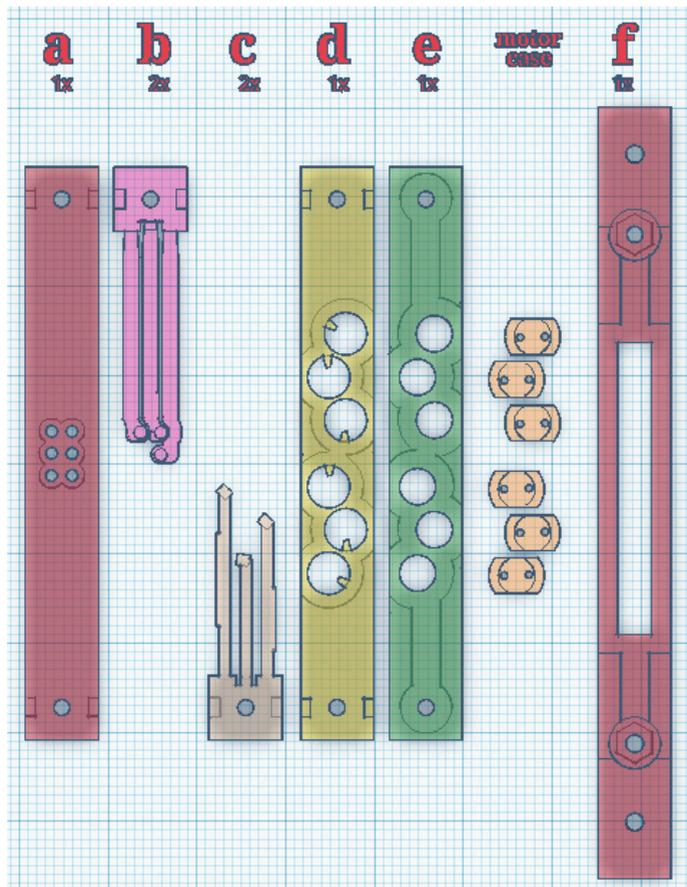


Figura 3.100 – Nomenclatura das peças a serem montadas.

1 - **Encaixe os motores sobre a peça “e”**. Vide a Figura 3.101 que ilustra a situação.



Figura 3.101 – Nomenclatura das peças a serem montadas.

2 - Depois **encaixe a peça “d” sobre eles**. É para ficar como ilustra a Figura 3.102.



Figura 3.102 – Motores encaixados.

3 - Em seguida, **monte todas as demais partes (“f”, “c”, “b” e “a”)** e insira os parafusos compridos nos lugares. É para ficar como ilustra a Figura 3.103.

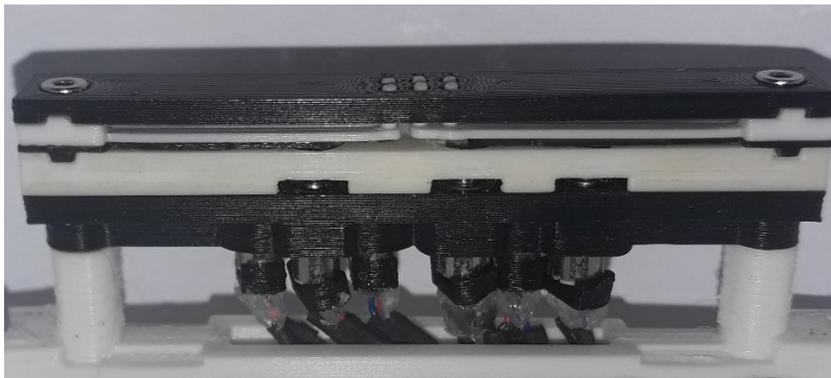


Figura 3.103 – Como a célula deve ficar montada.

Para entender melhor, assista ao vídeo *Funcionamento e montagem inicial da célula* disponibilizado no seguinte link:

https://www.youtube.com/watch?v=F_dQx-wSOiA.

4 - Com os parafusos pequenos (de 7mmX2,5mm) **fixe a célula ao encaixe adequado no box**, como ilustra a Figura 3.104. Cuidado para não danificar os fios ou a célula.



Figura 3.104 – Como a célula deve ficar no box.

5 - Novamente, com os parafusos pequenos, **fixe o Arduino ao box**, como mostra a Figura 3.105.

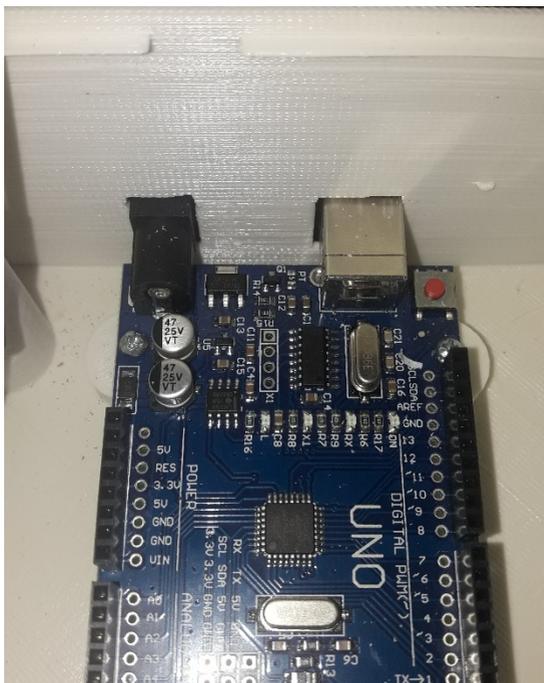


Figura 3.105 – Como o Arduino deve ficar no box.

Para visualizar este passo, assista ao vídeo *Fixando o Arduino no box* disponibilizado no seguinte link:

<<https://youtu.be/bYB3rNkiyQ>>.

6 - Para a **placa de controle** temos 2 possibilidades. No caso da placa ilhada basta encaixá-la sobre o Arduino (Figura 3.106).



Figura 3.106 – Como conectar a placa ilhada ao Arduino.

No caso da placa específica, a ideia é parafusá-la na lateral da caixa. Porém, o encaixe dependerá de um ajuste fino do tamanho dos fios. No nosso exemplo, optamos por deixar a placa solta. Os cabos fornecem um bom isolamento pela distância da placa do Arduino. Vide a Figura 3.107 para ver como ficará, ao final da montagem, a placa específica no box.



Figura 3.107 – Placa específica no box.

7 - Agora é o momento de **encaixar os cabos**. Faça isso com cuidado e conforme o esquema elétrico. Para os cabos dos motores, na realidade você não saberá se eles estão no sentido correto até que ligue o software do Celta. O procedimento é o seguinte:

No Arduino, carregue o software do Celta, como indicado no capítulo seguinte. Assim que o Arduino for carregado, os pinos da célula subirão e descerão. Então, verifique se algum deles está invertido. Então, neste caso, basta trocar os cabos do motor que está invertido. Por exemplo, se após você ligar o Celta os pontos 1 e 2 ficarem para cima (como na figura 3.108).

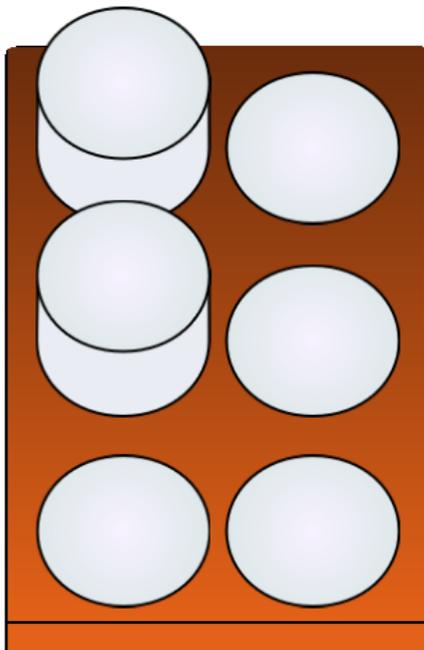


Figura 3.108 – Ilustração com pinos 1 e 2, indevidamente, para cima.

Neste caso você identificou que os motores 1 e 2 estão com a polaridade invertida. Então, lembre-se do Quadro 3.6 com o esquema das ligações dos motores. No motor 1 tínhamos o + no m1 e o - no m2. Então, basta inverter. Faça o mesmo com o motor 2. Só tenha muito cuidado, pois se ligar de forma errada poderá queimar os motores. Por exemplo, ligar coisas do motor 1 no motor 2 é algo que pode danificar tudo. Apenas inverta os cabos do mesmo motor!

Então, ligue novamente o Arduino e confira se ficou tudo certo, com os pontos todos para baixo após a subida inicial.

Para a placa específica, é necessário o encaixe cuidadoso dos fios. Inicie com os cabos dos motores, conforme ilustra a Figura 3.109.



Figura 3.109 – Conectando os cabos dos motores.

Depois, continue conectando os cabos das portas digitais, conforme mostra a Figura 3.100.



Figura 3.110 – Conectando os cabos das portas digitais.

Depois, conecte os cabos restantes, ou seja: dos botões (Figura 3.111), do GND (Figura 3.112) e do Vin (Figura 3.113).

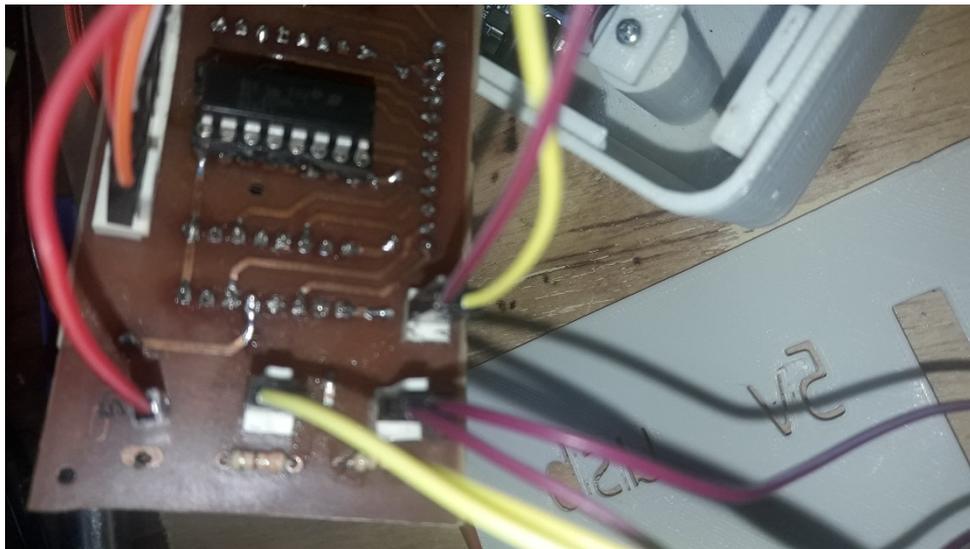


Figura 3.111 – Conectando os cabos dos botões.

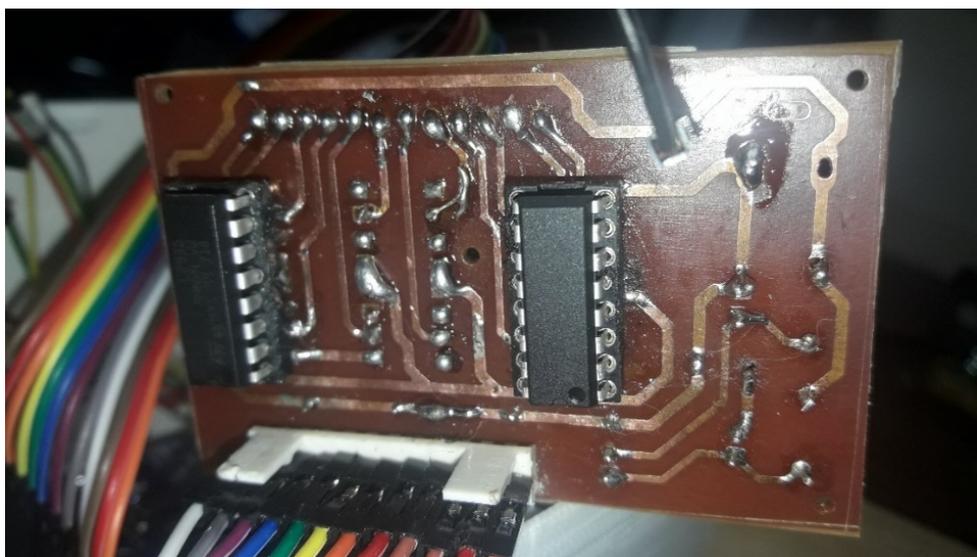


Figura 3.112 – Conectando o cabo do GND.

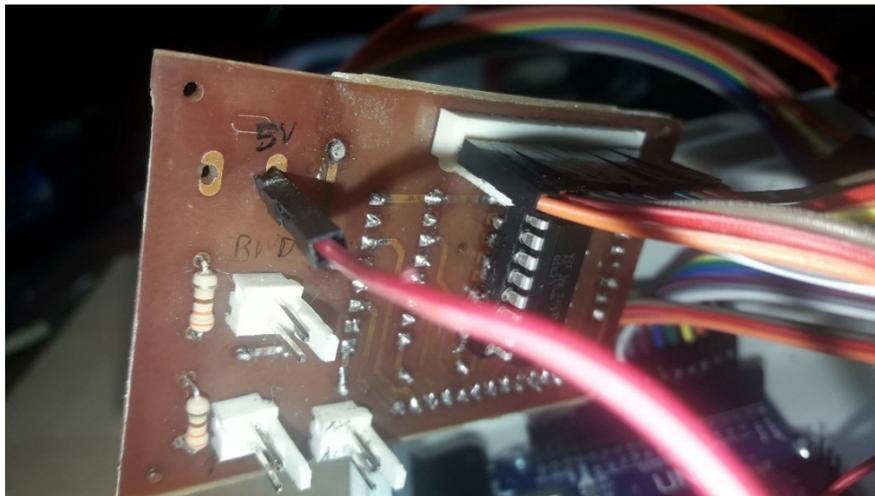


Figura 3.113 – Conectando o cabo do Vin.

Observe que, no final das contas, não utilizamos os conectores indicados e, sim, apenas os cabos Dupont, pois na montagem inicial devemos fazer diversos ajustes pois é importante poder alterar a ordem dos cabos. Somente após muitos ajustes e uso é que se aconselha fazer uma ligação mais fixa, incluindo a soldagem dos cabos no lugar de se usar conectores.

8 - Encaixe os **botões** na tampa do box (Figura 3.114) e depois faça a soldagem dos respectivos fios (Figurara 3.115) e conecte-os à placa de controle.



Figura 3.114 – Encaixando os botões na tampa do box.

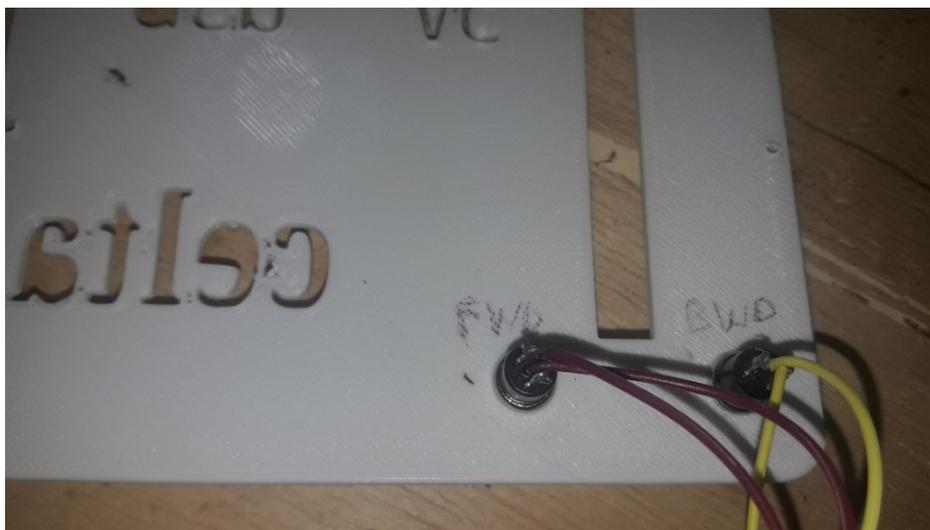


Figura 3.115 – Soldando os cabos dos botões na tampa do box.

9 - **Feche o box** com a tampa e parafuse-a com os parafusos pequenos. Lembre-se de usar apenas fontes DC de 5V em seu Celta. Vide as Figuras 3.116 e 3.117 para ver a aparência final do Celta montado.



Figura 3.116 – Visão da parte superior e frontal do Celta finalizado.



Figura 3.117 – Visão da parte traseira do Celta finalizado.

3.7 Resumo sobre a confecção e a montagem

Tudo o que foi dito neste capítulo é fundamental para o entendimento da confecção e montagem do Celta. Porém, indicamos aqui um resumo para ficar mais fácil de se ter uma visão geral do processo. Siga os passos:

1. Leia todo este manual para entender bem o processo.
2. Verifique e garanta os conhecimentos, pessoal e materiais necessários para construir um Celta.
3. Baixe os arquivos das modelagens.
4. Abra os arquivos no software fatiador, ajuste as peças e imprima a quantidade de peças necessárias.
5. Retire cuidadosamente os suportes das peças impressas.
6. Lime ou lixe as partes que não ficaram bem na impressão para que encaixem adequadamente.
7. Monte a placa de controle, ouilhada ou específica.
8. Prepare os motores.
9. Realize a montagem final de tudo.

Agora, basta partir para a parte do Software que é onde ocorreram os ajustes finais do hardware.

4. Software do Celta: instalação e utilização

Este capítulo, assim como os demais, apresenta algumas informações específicas para futuros desenvolvedores e para que se entenda totalmente o funcionamento do Celta, além dos conhecimentos mínimos para sua construção e utilização. Ou seja, não se preocupe com a complexidade de alguns temas aqui tratados, podem ser conhecimentos dispensáveis para uma abordagem mais prática na construção de um Celta.

Pois bem, neste capítulo abordaremos o Software. Na verdade, são dois softwares, um (chamado de *pontos.ino*) para ser carregado no Arduino, elaborado na linguagem do Arduino, e outro com interface gráfica, desenvolvido em C++ através do framework QT Creator (chamado de *Celta*), disponibilizado para a utilização em Windows.

Esta interface amigável exibe o conteúdo num bloco de texto, letra a letra, em braille, no dispositivo Celta. O programa percorre cada letra do conteúdo e envia o respectivo código da letra com os pontos que devem estar ativados ou desativados para que a letra esteja representada no formato braille. Por exemplo: se enviar o comando no formato "L1L2L3D4L5L6", todos os pontos exceto o quatro serão ativados.

A Figura 4.1 apresenta o aspecto da interface gráfica do Celta em sua versão para o sistema operacional Microsoft Windows. Na figura pode-se observar que acima das letras que seriam maiúsculas o software acrescenta um símbolo que será entendido pelo leitor do braille como sinal de maiúscula. E no caso dos números, o software também acrescenta o símbolo de número, seguindo a mesma lógica da leitura em braille.

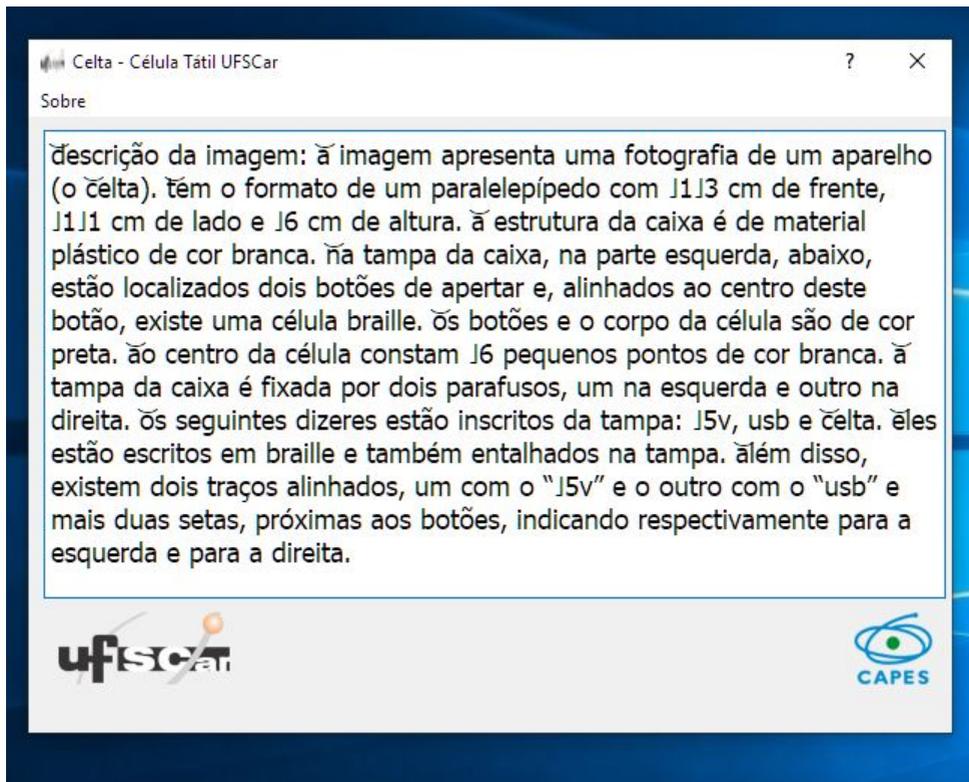


Figura 4.1 – Visualização do software Celta aberto no Windows.

Assista ao seguinte vídeo com uma breve demonstração dos pinos do Celta se movimentando para formar a palavra "celta":

<https://www.youtube.com/watch?v=Vb5_YXAUk2c>

4.1 Carregamento no Arduino

Para carregar o software do Arduino, é necessário seguir os passos:

1. Realizar a instalação da IDE do Arduino disponível no site: <https://www.arduino.cc/en/main/software>
2. Realizar o download do código fonte disponibilizado em um destes seguintes links:

https://github.com/juan-santos/celta_ufscar/blob/master/arduino/pontos/pontos.ino

ou

<https://celta.ufscar.br/wp-content/uploads/2020/04/pontos.zip>

3. Baixe o arquivo Bounce2-master.zip disponibilizado em um dos seguintes links:

https://github.com/juan-santos/celta_ufscar/blob/master/arduino/pontos/Bounce2-master.zip

ou

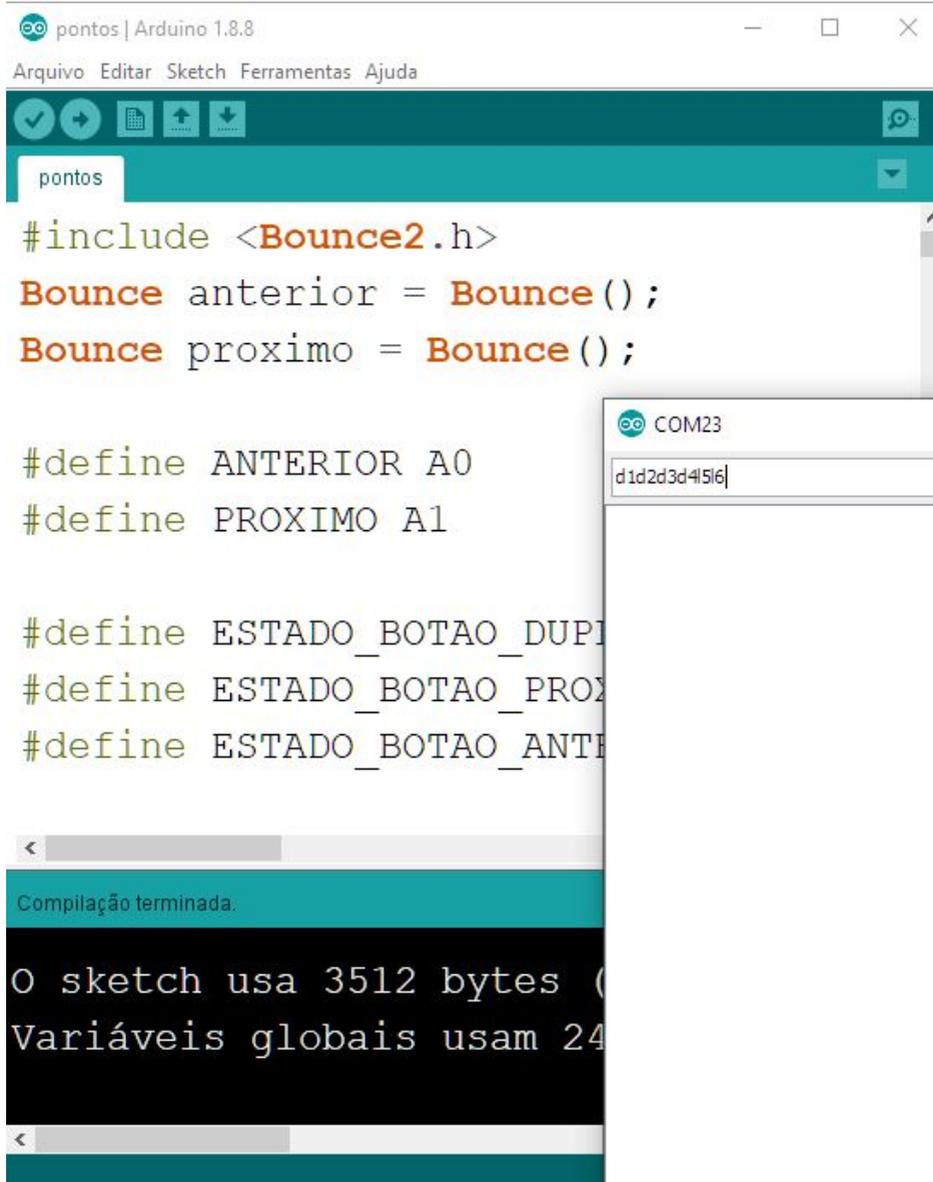
<https://celta.ufscar.br/wp-content/uploads/2020/04/Bounce2-master.zip>

Após baixados, os seguintes arquivos estarão em seu computador, conforme a Figura 4.2.

Nome	Data de modificaç...	Tipo	Tamanho
 Bounce2-master.zip	13/08/2019 08:35	Arquivo ZIP do Wi...	248 KB
 pontos.ino	13/08/2019 08:35	Arduino file	6 KB

Figura 4.2 – Arquivos necessários para a o software *pontos*.

4. Após o download, será necessário incluir o arquivo "Bounce2-master.zip" na biblioteca de sua IDE. Para isso, basta ir no menu "Sketch"->"incluir biblioteca"->"Adicionar biblioteca .ZIP"
5. Após realizar essa configuração, basta Abrir na IDE do Arduino o arquivo *pontos.ino*, conectar seu Arduino ao computador, escolher o modelo de placa correto (Arduino Uno) e a porta correta. Depois é só apertar no botão "carregar" para enviar o código ao aparelho. A Figura 4.3 mostra um pouco do software *pontos* na IDE do Arduino.



The image shows a screenshot of the Arduino IDE interface. The main window displays a sketch named "pontos" with the following code:

```
#include <Bounce2.h>
Bounce anterior = Bounce ();
Bounce proximo = Bounce ();

#define ANTERIOR A0
#define PROXIMO A1

#define ESTADO_BOTAO_DUPI
#define ESTADO_BOTAO_PROX
#define ESTADO_BOTAO_ANTE
```

Below the code editor, a status bar indicates "Compilação terminada." (Compilation finished). Below that, a black console window shows the following output:

```
O sketch usa 3512 bytes (
Variáveis globais usam 24
```

In the bottom right corner, a serial monitor window titled "COM23" is open, displaying the hexadecimal value "d1d2d3d4f5f6".

Figura 4.3 – Visão de trecho do software pontos na IDE.

4.2 Software leitor para o computador

Para a instalação da interface gráfica do software Celta, no Windows, basta baixar os arquivos de instalação para Windows neste link:

<<https://celta.ufscar.br/repositorio-de-arquivos/>>

Salve e extraia todos os arquivos em uma pasta nomeada como Celta, ou outro nome que desejar; e para executar o programa, basta clicar no arquivo *Celta.exe* que fica na pasta *bin* que aparecerá.

1. Após a conexão correta do Celta e a inicialização do software Celta, o texto que estiver contido em seu *clipboard* (área de transferência) será levado para a caixa de texto do software para a navegação inicial.
2. Para navegar entre as letras, basta utilizar os botões contidos no hardware do Arduino. Um botão serve para avançar enquanto o outro serve para voltar.
3. Para avançar/voltar uma letra, basta um clique rápido. Se o clique for um pouco mais longo, o software interpretará como avanço/retrocesso de toda uma palavra separada por espaço.
4. O clique simultâneo de ambos os botões será interpretado de forma que o texto volte ao início, enquanto que, se o clique for mais duradouro, o software interpretará para obter o texto contido na *clipboard* (área de transferência) daquele momento.
5. Caso o Arduino seja desconectado durante a execução do projeto, será necessário fechar o programa e abrir novamente. A configuração serial do Arduino com o PC é realizada ao iniciar o programa.

4.3 Detalhamento sobre o software do Arduino para desenvolvedores

Para desenvolvedores e curiosos, iremos apresentar um pouco da lógica utilizada para montar o software *pontos* que é carregado no Arduino.

O código inicia com a inclusão da biblioteca *Bounce2.h*:

```
#include <Bounce2.h>
Bounce anterior = Bounce();
Bounce proximo = Bounce();
```

Depois ocorrem diversas definições (*define*):

```
#define ANTERIOR A0
#define PROXIMO A1

#define ESTADO_BOTAO_DUPLO 1
#define ESTADO_BOTAO_PROXIMO 2
#define ESTADO_BOTAO_ANTERIOR 3

#define ESPERA_MEDIA 300

#define ON HIGH
#define OFF LOW

#define VAZIO_PADRAO B00000000

#define ATIVAR_1 B10000000
#define DESATIVAR_1 B01000000

#define ATIVAR_2 B00100000
#define DESATIVAR_2 B00010000

#define ATIVAR_3 B00001000
#define DESATIVAR_3 B00000100

#define ATIVAR_4 B00000010
#define DESATIVAR_4 B00000001

#define TEMPO_DELAY_LOOP 50
#define TEMPO_DELAY_MOTOR 400
#define TEMPO_DELAY_INICIAR 200
```

Então é apresentado o void inicializar. Ele objetiva fazer os pontos subirem e descerem assim que o Celta é ligado:

```
void inicializar(){
    PORTB = (ATIVAR_2 | ATIVAR_3 | ATIVAR_4);
    PORTD = (ATIVAR_1 | ATIVAR_2 | ATIVAR_3);
    delay(TEMPO_DELAY_INICIAR);

    PORTB = (DESATIVAR_2 | DESATIVAR_3 | DESATIVAR_4);
    PORTD = (DESATIVAR_1 | DESATIVAR_2 | DESATIVAR_3);
    delay(TEMPO_DELAY_INICIAR);
}
```

Depois é apresentado o void *executaAlteracoes*:

```
void executaAlteracoes(int b, int d){
    PORTB = b;
    PORTD = d;
}
```

Ainda é apresentado o void *desativarPontos*:

```
void desativarPontos(){
    PORTB = B00000000;
    PORTD = B00000000;
}
```

Então é iniciado o void setup, ele já contém alguns comentários que ajudam o programador a entender o que se deseja em cada parte do código:

```
void setup() {
    //configuro os pinos dos botões
    pinMode(ANTERIOR, INPUT_PULLUP);
    pinMode(PROXIMO, INPUT_PULLUP);

    //botão anterior
```

```
anterior.attach(ANTERIOR);  
anterior.interval(5); // interval in ms
```

```
//botão proximo  
proximo.attach(PROXIMO);  
proximo.interval(5); // interval in ms
```

```
DDRB = B11111111; //8 -> 13  
DDRD = DDRD | B11111100; //2 -> 7
```

```
//1 input, 0 output  
Serial.begin(9600);
```

```
inicializar();  
inicializar();
```

```
desativarPontos();  
}
```

Após o void setup é apresentado o void *atualizaBotoes*:

```
void atualizaBotoes(int *ant, int *prox){  
  anterior.update();  
  *ant = anterior.read();  
  
  proximo.update();  
  *prox = proximo.read();  
}
```

Finalmente temos o void loop, também comentado.

```
void loop() {  
  int executa = 0, ant = 0, prox = 0;  
  long tempoInicial = 0, tempoFinal = 0, estado = 0;  
  atualizaBotoes(&ant, &prox); //obtenho a situação atual dos  
  botões
```

```
if (ant == HIGH && prox == HIGH) { //se ambos os botões  
estiverem pressionados
```

```
while (ant == HIGH && prox == HIGH) { //enquanto os botões  
forem mantidos pressionados
```

```
if (executa == 0) {
```

```
tempoInicial = millis(); //tempo inicial
```

```
estado = ESTADO_BOTAO_DUPL0;
```

```
executa++;
```

```
}
```

```
atualizaBotoes(&ant, &prox); //obtenho o estado atual dos  
botões
```

```
//possível tempo final, caso o usuário pare de apertar os  
botões
```

```
tempoFinal = millis();
```

```
}
```

```
while (ant == HIGH || prox == HIGH) { //enquanto o usuário  
não soltar os dois botoes o programa nao enviara ou recebera  
mensagens
```

```
atualizaBotoes(&ant, &prox); //obtenho o estado atual dos  
botões
```

```
delay(50);
```

```
}
```

```
}
```

```
else {
```

```
if (ant == HIGH || prox == HIGH) { //se um dos botões estiver  
pressionados
```

```
while (ant == HIGH || prox == HIGH) { //enquanto o usuário  
não parar de pressionar todos os botões
```

```
    if (executa == 0) {
        tempoInicial = millis();
        if (ant == HIGH) {
            estado = ESTADO_BOTAO_ANTERIOR; //se apenas o botão
anterior estiver pressionado
        } else {
            estado = ESTADO_BOTAO_PROXIMO; //se apenas o botão
próximo estiver pressionado
        }
        executa++;
    }
    atualizaBotoes(&ant, &prox); //obtenho o estado atual dos
botões
    tempoFinal = millis(); //possível tempo final, caso o usuário
pare de apertar os botões
}
}
}
```

```
long tempoPressionado = tempoFinal - tempoInicial;
```

```
switch (estado) {
    case ESTADO_BOTAO_ANTERIOR:

        if (tempoPressionado < ESPERA_MEDIA) {
            Serial.write("-1");
        } else {
```

```
    Serial.write("--1");
}

break;
case ESTADO_BOTAO_PROXIMO:
    if (tempoPressionado < ESPERA_MEDIA) {
        Serial.write("+1");
    } else {
        Serial.write("++1");
    }
    break;
case ESTADO_BOTAO_DUPLO:
    if (tempoPressionado < ESPERA_MEDIA) {
        Serial.write("+0");
    } else {
        Serial.write("++0");
    }
    break;
}

if (Serial.available()) {
    int atualB = VAZIO_PADRAO;
    int atualD = VAZIO_PADRAO;

    while (Serial.available()) {
        char v1Funcao = Serial.read();
```

```
int vlPonto = Serial.parseInt();
  escolheSentido(vlPonto, vlFuncao, &atualB, &atualD);
}
```

```
executaAlteracoes(atualB, atualD);
Serial.write("ok");
```

```
delay(TEMPO_DELAY_MOTOR);
desativarPontos();
}
```

```
delay(TEMPO_DELAY_LOOP);
}
```

```
int escolheSentido(int porta, char operacao, int *sinalB, int
*sinalD) {
  if (operacao == 'I') {
    switch (porta) {
      //registradores b
      case 1:
        *sinalB = *sinalB | ATIVAR_2;
        break;
      case 2:
        *sinalB = *sinalB | ATIVAR_3;
        break;
      case 3:
```

```
    *sinalB = *sinalB | ATIVAR_4;
    break;
//registradores c
case 4:
    *sinalD = *sinalD | ATIVAR_1;
    break;
case 5:
    *sinalD = *sinalD | ATIVAR_2;
    break;
case 6:
    *sinalD = *sinalD | ATIVAR_3;
    break;
}
} else {
switch (porta) {
//registradores b
case 1:
    *sinalB = *sinalB | DESATIVAR_2;
    break;
case 2:
    *sinalB = *sinalB | DESATIVAR_3;
    break;
case 3:
    *sinalB = *sinalB | DESATIVAR_4;
    break;
//registradores c
```

```
case 4:
    *sinalD = *sinalD | DESATIVAR_1;
    break;
case 5:
    *sinalD = *sinalD | DESATIVAR_2;
    break;
case 6:
    *sinalD = *sinalD | DESATIVAR_3;
    break;
}
}
}
```

4.4 Instalação especial para desenvolvedores

Se você é desenvolvedor e deseja fazer aprimoramentos na interface gráfica do Celta, siga os seguintes passos:

1. Baixar o IDE QT Creator 4.8.1 - no projeto está sendo utilizado a versão 4.8.1 num ambiente com sistema operacional Ubuntu 18.04.3 LTS. O link é o seguinte:

<https://www.qt.io/download>

2. Realizar a instalação da IDE QT Creator.
3. Realizar o download do código fonte disponibilizado em um dos seguintes links:

https://github.com/juan-santos/celta_ufscar/tree/master/Celta

ou

<https://celta.ufscar.br/wp-content/uploads/2020/04/CeltaProjetoEmC.zip>

4. Para incluir o projeto no workspace do QT Creator, selecione o arquivo "Celta.pro" para que todos os arquivos sejam carregados no ambiente.
5. Conecte o Arduino ao computador antes de executar o projeto. Nesse momento, o hardware do projeto Celta já deve estar conectado nas portas corretas do Arduino.
6. Ao executar o projeto, o hardware será ajustado para que todos os pinos subam e desçam duas vezes. Verifique se todos os pinos estão seguindo a ação esperada, pois este é o momento para verificar se existe algum mau contato ou se as portas de um determinado pino estão invertidas.

5. Aprimoramentos e desdobramentos futuros para o projeto²¹

Durante a execução deste projeto, convivemos com os diversos desafios, conforme seguem:

1. Possibilidade de leitura em braille com ou sem deslocamento lateral do dedo. Vimos que é possível a leitura no Celta com ou sem o deslocamento lateral. Na verdade, o usuário fica à vontade para explorar a célula conforme desejar.
2. Possibilidade de leitura em braille com apenas um dedo ou com mais de um. Em nosso protótipo, utilizamos apenas uma célula, mas deixamos em aberta a possibilidade de futuros desenvolvimentos com mais de uma célula, já que esta tem um tamanho padrão e que podem ser montadas de forma modular. O senão é a necessidade de um aprimoramento no software e na placa de controle para suportar mais células.
3. A velocidade dos motores de baixo custo ser suficiente para um sistema que permita uma leitura veloz. Os motores se mostraram com uma velocidade suficiente para uma leitura razoável. O senão é que eles se mostraram não tão robustos em termos de força, ou seja, o usuário deve levantar o dedo da célula para que os pontos sejam sempre posicionados corretamente.
4. O tamanho do equipamento ser pequeno e de fácil transportabilidade. Neste aspecto, vimos que o tamanho e peso são adequados e de fácil portabilidade por usuários.
5. Possibilidade de deixar o sistema pronto para futuras implementações envolvendo wi-fi e bluetooth. Como ainda ficaram livres diversas portas do Arduino, são facilmente adaptáveis aprimoramentos com módulos de wi-fi ou bluetooth ao Celta.

²¹ Obs.: Partes do texto deste item já figuram em outras publicações dos autores relacionadas com este tema. A saber: SANTIAGO, Glauber Lúcio Alves. *Arduino para o ensino musical*. Ebook da disciplina Recursos tecnológicos para o ensino de música, Departamento de Artes e Comunicação, UFSCar, 2020, 65. p.

6. O uso de partes de fácil aquisição e/ou de partes utilizando impressoras 3D. Conseguimos executar o projeto utilizando peças em impressora 3D e componentes eletrônicos de fácil aquisição. De qualquer forma, fica o desafio para futuras equipes, em vez de utilizarem partes plásticas utilizarem partes metálicas no Celta. Imaginamos que com metal os ajustes sejam mais fáceis de serem implementados.
7. A fácil construção/replicação por terceiros. Pensamos que terceiros poderão construir de forma relativamente fácil o Celta. Isso dependerá do conhecimento, interesse e dedicação dos interessados. Porém, buscou-se, neste manual, explicar de forma bastante didática todos os passos para este empreendimento.
8. O software de reconhecimento de texto compatível com a maioria dos editores de textos padrão (formatos .doc, .txt, PDF etc.). O software do Celta lê textos, mas apenas os caracteres, as informações sobre imagens e outros gráficos não são lidos, assim como os demais leitores de tela a aplicações para cegos. Este conteúdo adicional deve ser provido por meio de descrições textuais de apoio ao cego.
9. A possibilidade de deixar o software de reconhecimento de texto compatível a leitores de tela em geral. Esta possibilidade não foi conseguida em termos absolutos. Na verdade, um leitor de tela lê o conteúdo da interface, porém o Celta não é um software instalado diretamente no sistema operacional, de forma a poder ser configurado como um leitor de tela. Porém, este aprimoramento pode ser desenvolvido futuramente.
10. O conforto na usabilidade do sistema. O sistema mostrou-se suficientemente confortável. Porém, obviamente um display braille profissional é mais confortável de ser utilizado, sobretudo por conta de ter mais células disponíveis.
11. A utilização de sons. Na proposta inicial havíamos pensado na necessidade de um sinalizador sonoro para indicar para o usuário o início da atividade do Celta, porém, verificamos que o próprio movimento dos motores já resulta em um sinal sonoro suficientemente audível. Para o futuro, outros aprimoramentos seriam a utilização de sintetizadores de voz ou a reprodução

de arquivos de áudio com informações adicionais ou paralelas à célula.

Palavras finais do coordenador do projeto

Este projeto foi bastante desafiador e sem a dedicação dos colaboradores desta equipe o projeto certamente não haveria auferido o mesmo êxito. Por isso, deixo aqui meus agradecimentos ao sr. Cristian Pendenza por haver abraçado esta causa logo nos primeiros momentos. Sem sua delicadeza e organização, o projeto não teria sido possível. Agradeço ainda ao sr. Juan Santos pelo brilhantismo técnico e interesse. Por fim, agradeço à srta. Clarissa Bengtson e ao sr. Douglas Pino pela gestão do projeto e pela parceria de sempre, em todos os momentos, ruins e bons.

Espero que este material de divulgação seja útil para futuros empreendimentos na área da acessibilidade no país. Desejo sucesso a todos.

Prof. Glauber Santiago 13/4/2020.

6. Referências

6.1 Referências de estudo

LEE, Jun Su; LUCYSZYN, Stepan. A micromachined refreshable Braille cell. **Journal of Microelectromechanical Systems**, v. 14, n. 4, p. 673-682, 2005.

GOCH, Bram. Optimizing a solenoid for a Robocup kicker. **DCT rapporten**, v. 2006, 2006.

MUBVUMA, Tinashe Harry. **Alternative Actuators for Braille Devices**. E90: Project Report. 2014.

SONG, Chang-Woo; LEE, Seung-Yop. Design of a solenoid actuator with a magnetic plunger for miniaturized segment robots. **Applied sciences**, v. 5, n. 3, p. 595-607, 2015.

DUARTE-BARÓN, Katherin et al. Design and construction of a device for facilitating the learning of Braille literacy system. **Ingeniería y competitividad**, v. 18, n. 1, p. 79-92, 2016.

HENLEY, Thomas; GUVENCH, Mustafa. Exploring affordable alternatives to refreshable braille displays. In: **2018 ASEE Northeast Section Conference**. 2018.

SCHIMPF, Paul H. A detailed explanation of solenoid force. **International Journal on Recent Trends in Engineering & Technology**, v. 8, n. 2, p. 7, 2013.

HUDECEK, Milan. **Braille display**. U.S. Patent n. 5,453,012, 26 set. 1995.

DATA Sheet do NCV7726B - Half-Bridge Driver - ON Semiconductor
<<https://www.onsemi.com/products/power-management/motor-drivers/motor-drivers-brushed/ncv7726b>

Dutra, Glênon. Eletromagnetismo. Apostila para a disciplina Física. <
<https://docplayer.com.br/38184720-Nome-professor-glenon-dutra.html>>

Nasson Pereira de Alcântara Junior e Claudio Vara de Aquino Apostila sobre Eletromagnetismo 1. UNESP
<https://www.passeidireto.com/arquivo/17365088/solenoide-e-indutancia>

6.2 Referências utilizadas no texto

BLIKSTEIN, P.; KRANNICH, D. The makers' movement and FabLabs in education: experiences, technologies, and research. In: international conference on interaction design and children, 12., 2013, Nova York. **Proceedings...** Nova York: ACM, 2013. p. 613-616.

BLUM, J. **Explorando o Arduino**: Técnicas e ferramentas para mágicas de engenharia. Rio de Janeiro: Alta books. 2016.

BRASIL. **Decreto Federal** nº 5.296, de 2 de dezembro de 2004.

BRASIL. **Lei nº 13.146**, de 06 de julho de 2015 – Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm. Acesso em: 14 de abr. de 2020.

BRASIL. **Lei nº 10.098**, de 19 de dezembro de 2000. Disponível em: http://www.planalto.gov.br/ccivil_03/LEIS/L10098.htm. Acesso em: 14 de abr. de 2020.

CAMPBELL, T.; WILLIAMS, C.; IVANOVA, O.; GARRETT, B. **Could 3D Printing Change the World?** Technologies, Potential, and Implications of Additive Manufacturing. Washington: Atlantic Council, 2011.

EKVALL, S.; HÖGLUND, C. e KARLSSON, A. **Iniciação à Leitura do Braille**. In: Sobre a Deficiência Visual. Site <http://www.deficienciavisual.pt/txt-iniciacao.htm> 13/04/2020.

HATCH, M. **The Maker Movement Manifesto**: Rules for Innovation in the New World of Crafters, Hackers, and Tinkerers. Nova York: McGrawHill Education, 2013. 213 p.

HULL, C. W. The birth of 3D printing. **Research-Technology Management**, v. 58, n. 6, p. 25-30, 2015.

INSTRUCTOR BRAILLE. **El código Braille**. 2018. Disponível em: <<http://instructorbraille.blogspot.com.br/p/el-codigo-braille.html>>. Acesso em: 17 out. 2019.

MIRADOR INTERNACIONAL. **São Paulo**: Encyclopaedia Britannica do Brasil, 1987.

MONK, S. **30 Projetos com Arduino**. Porto Alegre: Bookman, 2014.

SANTIAGO, G. L. A.; BENGTON, C.; PINO, D.; PENDENZA, C; KAWAKAMI, C. **Célula tátil Braille**: Desenvolvimento de uma Ferramenta da Acessibilidade. In: Anais do ESUD/CIESUD, 2019.

SANTIAGO, G. L. A.; OLIVEIRA, C. D. Movimento maker e IoT para educação musical: possibilidades com Impressão 3D, Software Processing e Arduino. In:

DIAS-TRINDADE, S.; MILL, D. (Orgs.). **Educação e Humanidades Digitais: Aprendizagens, Tecnologias e Cibercultura**. Coimbra: Imprensa da Universidade de Coimbra, 2019. p. 285-308.

SANTIAGO, G. L. A.; OLIVEIRA, C., D. **Introdução à modelagem e impressão**. São Carlos: Edutec (curso de especialização em Educação e tecnologias), 2020, 35 p.

SANTIAGO, G. L. A. **Arduino para o ensino musical**. Ebook da disciplina Recursos tecnológicos para o ensino de música, Departamento de Artes e Comunicação, UFSCar, 2020, 65. p.

SKORTZARU, A. **O impacto da impressão 3D e da cultura maker na educação**. iMasters, 18 abr. 2017. Disponível em: <<https://imasters.com.br/desenvolvimento/o-impacto-da-impressao-3d-e-da-cultura-maker-na-educacao>>. Acesso em: 27 set. 2019.

VISHBOX. **Entenda o impacto do movimento maker e da impressão 3D**. 10 out. 2019. Disponível em: <<https://www.wishbox.net.br/entenda-o-impacto-do-movimento-maker-e-da-impressao-3d/#btn-continuar-lendo>>. Acesso em: 27 set. 2019.

Apêndice: Repositório de Arquivos

Baixe aqui os arquivos disponibilizados para a realização do projeto

[Ebook CELTA: Sistema de célula tátil para leitura Braille – Manual de confecção:](#)

- Em formato PDF
- Em formato Epub2

[Exercícios com o Celta \(Referentes ao capítulo 2\)](#)

[Arquivos com os Parâmetros de impressão para o Simplify3D](#)

[Arquivos com a Modelagem para a impressão 3D](#)

[Esquemas guia para a placa ilhada](#)

[Arquivos com a Máscara da PCI para impressão](#)

[Biblioteca Bounce2-master para a IDE do Arduino](#)

[Sketch Pontos para o Arduino](#)

[Celta 1.0 para Windows](#)

[Projeto aberto do software Celta em C++](#)

Links para material do projeto em sites externos

Prática *Virtual Celta – Treinamento Braille*

<<https://scratch.mit.edu/projects/380045807/>>

Projeto CAD da célula

<<https://www.tinkercad.com/things/jgzAcNGjWxt>>

Projeto CAD do box

<<https://www.tinkercad.com/things/iValvIHpQ6w>>

Simulação CELTA: L293D com 1 motor girando

<<https://www.tinkercad.com/things/3wRF7YdbZxw>>

Simulação CELTA: Circuito simulado

<<https://www.tinkercad.com/things/6iX4XzukzZ2>>

Links no Github

<https://github.com/juan-santos/celta_ufscar/blob/master/arduino/pontos/pontos.ino>

<https://github.com/juan-santos/celta_ufscar/blob/master/arduino/pontos/Bounce2-master.zip>

Links para os vídeos do projeto em alta definição

1.6.3- Protótipo 1 em funcionamento

<<https://www.youtube.com/watch?v=UYjkagUZ6Yg>>

1.6.4-Protótipo em uso

<<https://www.youtube.com/watch?v=snuEjvb7nf4>>

3.1-Apresentação

<<https://www.youtube.com/watch?v=29Zvcs4ppus>>

3.2.1-Projeto CAD da Célula:

<<https://www.youtube.com/watch?v=LAN8gASXNF8>>

3.2.2-Configuracoes do Fatiador para a Célula

<<https://www.youtube.com/watch?v=LSDsrDxbaGs>>

3.2.3-Impressao da Célula

<<https://www.youtube.com/watch?v=r3T2u6w8BYM>>

3.3.1-Projeto CAD do box

<<https://www.youtube.com/watch?v=WsjhkZZW78w>>

3.3.1- Impressão e retirada de suporte do protótipo

< <https://youtu.be/wmgU2HlmxTE>>

3.3.1-Retirada dos suportes do box

<<https://youtu.be/Yx9LkpKjvsA>>

3.3.2-Configurações do Fatiador e de Impressão do Box

<<https://www.youtube.com/watch?v=8GBzCLLChCI>>

3.4.4-Projeto Diagrama elétrico e confecção da placa de controle

<<https://www.youtube.com/watch?v=QGipBsTa0vI>>

3.6-Funcionamento e montagem inicial da célula

<https://www.youtube.com/watch?v=F_dQx-wSOiA>

3.6-Fixando o Arduino no box

<<https://youtu.be/bYB3rNkiyjQ>>

4. Celta Escrevendo celta celta

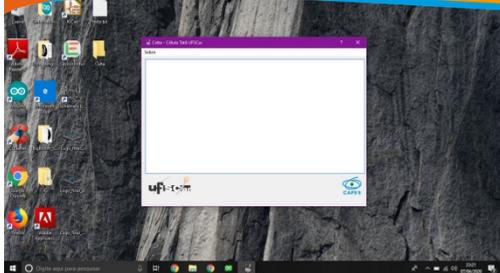
<https://www.youtube.com/watch?v=Vb5_YXAUk2c>



CELTA: Sistema de célula tátil para leitura Braille



Site completo com instruções e materiais

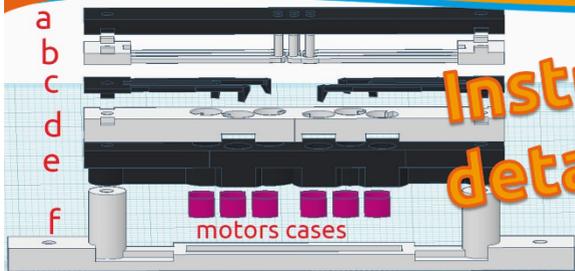


Software



Hardware





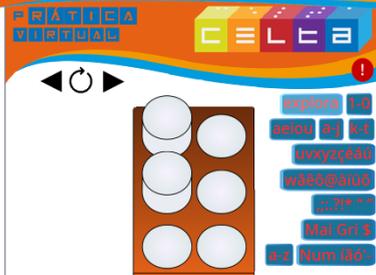
Instruções detalhadas



Diversos vídeos explicativos



Elabore o seu



Versão on-line para treinamento





UNIVERSIDADE FEDERAL
DE SÃO CARLOS



FINANCIAMENTO

