

Desenvolvimento de Tecnologias e de Funcionalidades Avançadas de Impressão 3D

Sara Rodriguez

RELATÓRIO TÉCNICO CIENTÍFICO APRESENTADO À
FUNDAÇÃO DE AMPARO À PESQUISA DO ESTADO DE SÃO PAULO -
(FAPESP)

Processo FAPESP nº: 2013/12374-2

Período: 01/08/2013 a 31/12/2013

Responsável: Felipe Alves Moura

Durante o desenvolvimento deste trabalho a autora recebeu auxílio financeiro da FAPESP

Desenvolvimento de Tecnologias e de Funcionalidades Avançadas de Impressão 3D

Esta é a versão original do relatório técnico
elaborado pela bolsista Sara Rodriguez,
tal como submetido à FAPESP.

Agradecimentos

Agradeço primeiramente à FAPESP, pela concessão da minha bolsa de treinamento técnico, tipo TT4A, e ter colaborado para que eu atuasse na Metamáquina. Graças a isso adquiri muito conhecimento nesta área tão inovadora, conheci pessoas tão admiráveis e inteligentes e ainda obtive auxílio financeiro em uma etapa importante da minha vida. Gostaria muito de continuar contribuindo nesta área.

Agradeço a todos os meus colegas da Metamáquina:

Agradeço ao Filipe Moura, por ter confiado na minha capacidade perante o desafio de construir uma impressora com extrusão dupla, ter trocado o tipo da minha bolsa e ter insistido para que eu entrasse neste projeto.

Agradeço ao Felipe Sanches, por me ensinar a trabalhar com os *softwares*: *OpenSCAD*, de desenho tridimensional; *Arduino IDE*, de eletrônica e o *Pronterface*, de interface para a impressão 3D.

Agradeço ao Rodrigo Rodrigues da Silva, pelos seus ensinamentos e por ter-me fornecido dicas tão preciosas em diversos momentos oportunos.

Quero agradecer à minha mãe, que primou pelos meus estudos e pela minha carreira e sempre esteve presente em minha vida.

Agradeço, por fim, a Deus, que fez com que eu vivenciasse trabalhar por um semestre, de um ano muito complicado para mim e para a área em que eu atuava, em uma empresa tão incrível e inovadora. Espero utilizar o conhecimento adquirido na *Metamáquina* futuramente.

Resumo

RODRIGUEZ, S. **Desenvolvimento de Tecnologias e de Funcionalidades Avançadas de Impressão 3D.** 2013. 44 f. Relatório Técnico - FAPESP, Metamáquina, São Paulo, 2013.

A impressão tridimensional abordada neste relatório consiste em uma tecnologia de fabricação onde um modelo é elaborado através do depósito de sucessivas camadas de material plástico. Este processo tem sido cada vez mais ágil, flexível, acessível e mais interativo com o usuário.

A prototipagem rápida abrange um número crescente de áreas do conhecimento e é capaz de elaborar peças de diversos materiais de complexidade variada.

Neste relatório técnico são apresentados projetos, a fim de aprimorar o modelo atual da *Metamáquina 2*, tais como a elaboração de réguas de calibração e de um teste para a verificação de barras rosadas deste equipamento.

Também são apresentados os resultados de diversas bancadas de testes, tais como a de averiguação de temperatura e a de vibração. Inclusive, com foco na qualidade da impressão, foram realizados testes de extrusão e retração e de impressão com filamentos.

Também é apresentado um modelo de suporte de filamento elaborado com *software OpensCAD* e com processo de corte à laser.

A versatilidade da utilização de tecnologia de dupla extrusão é relevante, com ela é possível utilizar materiais dissolvíveis para imprimir peças que necessitam sustentação. É possível, também, elaborar peças de duas cores ou dois materiais distintos.

Por fim, a partir deste cenário, este relatório técnico apresenta o projeto de elaboração de um protótipo de impressora tridimensional com extrusão dupla, sendo apresentados os dados técnicos de *hardware* e *software* e impressões elaboradas.

Desta forma, este relatório apresenta aprimoramentos implementados e futuros na área de impressão tridimensional.

Palavras-chave: Prototipagem rápida, Desenvolvimento de Produtos, Impressão Tridimensional, Produção em Lotes Pequenos, Projeto de Máquinas.

Abstract

RODRIGUEZ, S. **Development of Technology and Advanced Features for 3D Printing.** 2013. 44 f. Technical Report - FAPESP, Metamáquina, São Paulo, 2013.

The three-dimensional printing presented in this report consists in a manufacturing technology which a model is maiden by depositing successive layers of plastic material. This process has been increasingly agile, flexible, accessible and more interactive with the user.

Rapid prototyping encompasses a growing number of areas of knowledge and it became possible to elaborate pieces of different materials and complexity.

This technical report presents projects in order to enhance the current printer model called *Metamáquina 2*, such as the elaboration of projects of calibration rules and a test to check threaded bars of this equipment.

Also are presented the results of some test stands, such as the temperature measurement and the vibration one. Even focusing on the quality of printing, extrusion and retraction and print tests with filaments were performed.

Also is presented a filament spool holder project maiden by the *software OpensCAD* by laser cutting process.

The versatility of using the double extrusion technology is relevant, it is possible to use a dissolvable material for printing parts that need support. You can also build pieces of two colors or two different materials.

Finally, this report presents the development of a project of a three-dimensional printer with dual extrusion, the technical data of *hardware* and *software* and the objects printed.

Thus, this report presents implemented and future improvements in the area of ??three-dimensional printing.

Keywords: Rapid Prototyping, Product Development, 3D Printing, Small Batch Production, Machine Design .

Sumário

Lista de Abreviaturas	ix
Lista de Símbolos	xi
Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Considerações Preliminares	1
1.2 Histórico	2
1.3 Tecnologias de Impressão Existentes	4
1.4 Condição Atual	5
1.5 Objetivos	6
1.6 Contribuições	6
1.7 Organização do Trabalho	6
2 Projetos Realizados	7
2.1 Réguas de Calibração	7
2.2 Teste de Barras	8
2.3 Porta Rolo de Filamento	9
2.4 Impressora 3D com extrusão dupla	10
2.4.1 Hardware	10
2.4.2 Software	12
2.4.3 Impressões Realizadas	18
3 Testes Realizados	23
3.1 Temperatura	23
3.1.1 Conceitos	23
3.1.2 Experimentos	24
3.2 Vibração	25
3.3 Extrusão e Retração	31
3.4 Teste de Filamentos	32
4 Conclusões	43
4.1 Considerações Finais	43
4.2 Sugestões para Pesquisas Futuras	44

A Scripts dos Desenhos do OpenSCAD	45
A.1 Porta Rolo de Filamento	45
A.2 Teste de Barras	49
A.3 Réguas de Calibração	57
A.4 Prendedor de Extrusores	60
B Scripts do Software Arduino IDE	63
B.1 Teste de Barras	63
B.1.1 Rotação Unidirecional e Constante	63
B.1.2 Rotação Bidirecional e Gradual	64
B.2 Arquivo configuration.h do Repetier-Firmware Adaptado para Dupla Extrusão	65
B.3 Teste de Temperaturas	88
B.4 Teste de Vibração	89
B.4.1 Gcode	89
B.4.2 Printcore	89
B.5 Teste de Extrusão	90
Referências Bibliográficas	91

Lista de Abreviaturas

3D três dimensões

ABS acrilonitrilo-butadieno-estireno (*Acrylonitrile Butadiene Styrene*)

CNC Controle Numérico Computadorizado (*Computer Numeric Control*)

DMLS Sinterização Seletiva de Metal por Laser (*Direct Metal Laser Sintering*)

FDM Fused Deposition Modeling

HIPS Poliestireno de Alto Impacto (*High Impact Polystyrene*)

Id Número Identificador

MDF Madeira de Média Densidade (*Medium-density fibreboard*)

MJM Multi-jato de Modelagem (*multi-jet modelling*)

ND Não Disponível

NTC Coeficiente Negativo de Temperatura (*Negative Temperature Coefficient*)

PLA ácido poliláctico (*Polylactic acid or polylactide*)

PTFE Politetrafluoretileno

PVA Álcool Polivinílico (*Polyvinyl alcohol*)

SHS Sinterização Seletiva por Calor (*Selective Heat Sintering*)

SLA Aparelho de Estereolitografia (*Stereo Litography Apparatus*)

SLM Fusão Seletiva a Laser (*Selective Laser Melting*)

SLS Sinterização a Laser (*Selective Laser Sintering*)

STL *STereoLitography ou Standard Tessellation Language*

UV Ultravioleta

Listas de Símbolos

β	Fator Beta do Termistor	[adm]
a	Fator de <i>Steinhart-Hart</i>	[adm]
b	Fator de <i>Steinhart-Hart</i>	[adm]
c	Fator de <i>Steinhart-Hart</i>	[adm]
D_{nom}	Diâmetro Nominal	[mm]
$Mult$	Quantidade de filamento extrudado em relação ao padrão	[%]
R	Resistência	[Ω]
R_0	Resistência a $25^{\circ}C$	[Ω]
$RawADC$	Valor de corrente lido pela placa	[A]
Ret	Comprimento de retração durante a impressão	[mm]
s	Desvio Padrão	[adm]
T	Temperatura	[K]
T_{fab}	Temperatura de Impressão Fornecida pelo Fabricante	[$^{\circ}C$]
T_{tb}	Temperatura do Extrusor para Teste de Impressão	[$^{\circ}C$]
T_{te}	Temperatura do Extrusor para Teste de Impressão	[$^{\circ}C$]
v_{max}	Velocidade máxima de extrusão	[mm/s]
\bar{x}	Média de Diâmetros Mensurados	[mm]

Lista de Figuras

2.1	Barras lisas a serem posicionadas com precisão	7
2.2	Régua de calibração frontal.	8
2.3	Régua de calibração dorsal.	8
2.4	Faces para o corte a laser das réguas de calibração.	8
2.5	Modelo para o equipamento de teste de barras.	9
2.6	Modelo elaborado para o porta rolo de filamento.	10
2.7	Faces para o corte a laser do porta rolo de filamento.	10
2.8	Protótipo elaborado para o porta rolo de filamento.	11
2.9	<i>Kit</i> de um dos extrusores preso na carcaça da impressora três dimensões (3D).	11
2.10	<i>Kit</i> de um dos extrusores preso na carcaça da impressora 3D.	12
2.11	Primeira versão do prendedor dos bicos extrusores.	12
2.12	Versão final do prendedor dos bicos extrusores - vista superior.	13
2.13	Versão final do modelo do prendedor dos bicos extrusores - visão 3D.	13
2.14	Protótipo do prendedor dos bicos extrusores.	14
2.15	Protótipo da impressora 3D.	14
2.16	Interface do <i>software Arduíno</i>	15
2.17	Interface do <i>Conversão de arquivos STereoLithography ou Standard Tessellation Language (STL) para .gcode pelo Pronterface</i>	16
2.18	Tela geral do <i>Slic3r</i>	17
2.19	Tela do <i>Slic3r</i> de propriedades de impressão.	17
2.20	Tela do <i>Slic3r</i> sobre propriedades do filamento.	17
2.21	Tela do <i>Slic3r</i> sobre propriedades da impressora.	18
2.22	Tela do <i>Pronterface</i>	19
2.23	Primeira peça impressa.	19
2.24	Impressão em acrilonitrilo-butadieno-estireno (<i>Acrylonitrile Butadiene Styrene</i>) (ABS) em duas cores.	20
2.25	Impressão em ABS e ácido poliláctico (<i>Polylactic acid or polylactide</i>) (PLA).	20
3.1	Termistores de vidro da <i>addtherm</i>	24
3.2	Desenho esquemático de um dos termistores da placa <i>Rambo</i>	25
3.3	Teste de termistores.	26
3.4	Média dos resultados dos ensaios para os três lotes de termistores.	26
3.5	Impressora Utilizada para o Teste de Vibração.	27
3.6	Disposição dos principais elementos testados.	28
3.7	Parafuso com porca com grande quantidade de cola trava-rosca <i>Loctite 243</i>	28
3.8	Parafuso com porca com pouca quantidade de cola trava-rosca <i>Loctite 243</i>	29

3.9	Parafusos com porca com cola trava-rosca <i>Tek Bond 128</i> - identificados em vermelho.	29
3.10	Parafuso com porca comum.	30
3.11	Parafuso com porca comum.	30
3.12	<i>T-slot</i> justo para a porca auto travante da empresa <i>seemeCNC</i> [see13].	30
3.13	Equipamento utilizado para o teste de extrusão adaptado para uso de tubo <i>Bowden</i> .	31
3.14	Detalhe do ensaio de extrusão com tubo <i>Bowden</i> .	32
3.15	Procedimento de impressão para filamento de 3,00mm.	33
3.16	Peça final impressa.	33
3.17	Tubo conversor para utilizar filamentos de 1,75mm.	34
3.18	Limpeza do bico extrusor.	34
3.19	Inserção do tubo conversor.	35
3.20	Visualização final da colocação do tubo conversor.	35
3.21	Procedimento de impressão para filamento de 1,75mm.	36
3.22	Peça final impressa.	36
3.23	Teste com filamento PLA da marca <i>Esun</i> cor azul claro.	37
3.24	Teste com filamento PLA translúcido da marca <i>Esun</i> cor azul.	37
3.25	Teste com filamento PLA da marca <i>Esun</i> cor cinza.	37
3.26	Teste com filamento PLA da marca <i>Esun</i> cor preta.	39
3.27	Teste com filamento PLA translúcido da marca <i>Esun</i> cor verde.	39
3.28	Teste com filamento Poliestireno de Alto Impacto (<i>High Impact Polystyrene</i>) (HIPS) da marca <i>Esun</i> cor natural.	39
3.29	Teste com filamentos PLA da marca <i>Macomp</i> .	40
3.30	Impressão com filamento Álcool Polivinílico (<i>Polyvinyl alcohol</i>) (PVA) da marca <i>Esun</i> .	40
3.31	Teste com filamento PVA da marca <i>Esun</i> .	41

Lista de Tabelas

3.1 Verificação do Diâmetro dos Filamentos	38
3.2 Perfis de fatiamento	38

Capítulo 1

Introdução

A impressão **3D** abordada neste relatório consiste em uma tecnologia onde objetos são obtidos através da deposição de camadas de material em um processo que usualmente demora algumas horas.

Esta tecnologia possibilita construir um objeto sólido a partir de um modelo computacional, em diversos materiais, a ponto de se fabricar alimentos, brinquedos, obras de arte, joias, órgãos, dentre outros.

Esta tecnologia, utilizada antes como prototipagem rápida na área da engenharia invadiu diversos ramos do conhecimento e tem-se mostrado mais importante e cada vez mais inovadora para o futuro próximo.

Neste capítulo será apresentado um breve histórico sobre a impressão **3D**, as tecnologias desenvolvidas e utilizadas atualmente e o seu funcionamento.

Serão apresentadas as áreas de atuação desta tecnologia que têm ganhado destaque.

Por fim, é apresentado o objetivo deste relatório técnico e suas principais contribuições neste ramo de atuação.

1.1 Considerações Preliminares

Este relatório é fruto do trabalho e pesquisa realizada pela autora durante a sua vigência na *Metamáquina*. Portanto, todos os dados obtidos, equipamentos utilizados, e produtos elaborados farão parte da contingência desta empresa e para a inovação de seus produtos.

Todos os *softwares* utilizados para a realização deste relatório são livres, de código aberto e estão disponibilizados na internet.

Assim que consolidada a nova impressora com extrusão dupla, o seu projeto será disponibilizado no repositório do *Github* da *Metamáquina* na internet, pois também segue a filosofia de utilização livre e aberta.

O *software* livre e o código aberto permitem a distribuição de conhecimento para todos e a livre utilização pelo usuário. Também é possível o seu aprimoramento através de usuários e desenvolvedores que tenham interesse neste ramo.

Desta forma, torna-se uma forma de aprimoramento, melhoria de custo-benefício e superação de desafios para o setor de impressão **3D**.

1.2 Histórico

Em 1860, *François Willème* foi capaz de capturar um objeto em **3D** utilizando câmeras ao seu redor, criando a fotoescultura [Stu11].

Em 1892, *Blanther* propôs um método de fatiamento para a produção de mapas topográficos [Woz09].

Em 1972, *Mastubara*, pertencente à *Mitsubishi Motors*, propôs a utilização de fotopolímeros a fim de elaborar camadas de objetos [Bea01].

Em 1981, *Hideo Kodama*, estudioso pertencente à *Nagoya Municipal Industrial Research Institute* publicou o primeiro relato a despeito de um sistema de prototipagem rápida utilizando fotopolímero [Ros09].

Em 1984, *Charles Hull*, fundador da empresa *3D Systems*, foi o responsável pela criação do Aparelho de Estereolitografia (*Stereo Litography Apparatus*) (**SLA**), que foi patenteado em 1987. Esta tecnologia permitiu elaborar um modelo **3D** a partir da atuação do laser em um fotopolímero [Sch13].

Em 1991, a *Stratasys*¹ produziu o primeiro equipamento Fused Deposition Modeling (**FDM**) [Str92]. Esse método utiliza polímero e um bico extrusor, que deposita camadas plásticas sob uma chapa aquecida. Essa tecnologia foi abordada nos experimentos descritos neste relatório técnico.

Em 1992, a empresa *3D Systems* elaborou a primeira máquina de impressão **3D SLA** [Hil12].

Neste mesmo ano a *DTM* produziu a primeira máquina Sinterização a Laser (*Selective Laser Sintering*) (**SLS**), que possui tecnologia similar à **SLA**, porém utiliza laser para fundir pó ao invés de solidificar líquidos poliméricos [D.11].

Em 1994, foi lançada a impressora de cera pela empresa *ModelMaker* [TW12].

Em 1997, a empresa *Aeromet* inventou a técnica de manufatura por laser aditivo [LC09].

Em 1999, cientistas foram capazes de utilizar a impressão **3D** para elaborar um suporte para permitir o crescimento de células de pacientes, a fim de gerar novos órgãos a serem transplantados [Hsu13].

No ano 2000 foi obtida a primeira impressora jato tinta **3D**, produzida pela empresa *Object Geometries*².

Neste mesmo ano foi elaborada a primeira impressora **3D** multicolorida, através da empresa *Z Corporation* [Cor95].

Em 2001, foi lançada ao mercado a primeira impressora **3D** compacta pela empresa *Solidimension* [TG01].

Em 2002, foi impresso um rim humano em miniatura pelo *Wake Forest Institute for Regenerative Medicine*. Os cientistas almejavam imprimir órgãos totalmente funcionais e implantáveis. Este

¹Maiores informações sobre a Stratasys podem ser encontradas no sítio: <http://www.stratasys.com>

²Corresponde à patente americana denominada *Apparatus and method for raised and special effects printing using inkjet technology*, de número 6644763

evento e outros importantes na área biomédica são apresentados na referência [Tor13].

Em 2005, o projeto *Reprap* emergiu graças ao *Dr. Adrian Bowyer*, da *University of Bath*. Este projeto foi inovador dado ao fato de ser elaborado em código aberto e *software* livre, incentivando a utilização e desenvolvimento do ramo de impressão 3D [Bow13].

Em 2008, o ramo da impressão 3D teve grandes descobertas:

Surgiu outra característica marcante do projeto da *Reprap*, a capacidade da própria impressora produzir, em grande quantidade, muitos de seus componentes estruturais.

A empresa *Stratasys* produziu o primeiro sistema FDM biocompatível [Str08].

A empresa *Shapeways* elaborou um sítio de venda de modelos a serem impressos em 3D [Sha13].

A empresa *Makerbot* produziu um sítio de compartilhamento de modelos gratuitos e pagos para impressão 3D [Mak13].

Em 2009, também ocorreram avanços importantes:

A empresa *Makerbot* produziu um kit para montar uma impressora 3D para o público em geral [Mak13].

A empresa *Organovo* elaborou a primeira impressão de um vaso sanguíneo através de impressão 3D [3DP12].

Neste mesmo ano, *Cody Wilson*, da empresa *Defense Distributed*, produziu um modelo para impressão de uma arma em 3D [Bar13c].

Em 2011, produziu-se o primeiro carro através do uso de impressão 3D, denominado *Urbee* através da empresa *Kor Ecologic*. Através do sítio [Eco13], pode-se encontrar o segundo modelo deste carro.

No ano de 2012 ocorreram outros desenvolvimentos interessantes neste ramo:

Foi elaborado o primeiro implante de mandíbula, produzido através do uso de impressão 3D na Holanda pela empresa *Layerwise* [Gri12].

Um rim elaborado com impressão 3D foi projetado e implantado em um paciente pelo cirurgião *Anthony Atala* [Ata13]. Este médico tem sido um dos responsáveis por incríveis avanços na área médica utilizando-se a impressão de células do próprio paciente para reconstrução de tecidos e órgãos.

Em 2013, ano atual deste relatório, esta área encontra-se em pleno desenvolvimento e tem se destacado em todos os meios de comunicação:

Foi realizado um implante parcial de rosto no paciente *Eric Moger* através de impressão 3D [Fei13].

O sítio da *Defense Distributed* foi obrigado a retirar os projetos da primeira arma impressa em 3D do ar [Dis13].

A própria empresa *Metamáquina* tem aparecido recorrentemente em diversos meios de comunicação.

1.3 Tecnologias de Impressão Existentes

A impressão estereolitográfica (também conhecida por **SLA**), corresponde a uma cuba com uma plataforma perfurada posicionada um pouco abaixo do nível superficial de polímero líquido fotocurável. Um feixe de laser Ultravioleta (**UV**) desenha a primeira fatia de um objeto na superfície do líquido, fazendo com que uma camada muito fina de fotopolímero endureça.

A plataforma é então abaixada e outra fatia é desenhada e endurecida pelo laser. Outra fatia é então criada, e assim sucessivamente, até que o objeto seja impresso e possa ser removido da cuba de fotopolímero, drenando o excesso de líquido, e curado.

Impressoras estereolitográficas são consideradas um dos tipos mais precisos de fabricação de objetos em **3D**, com espessura por camada de apenas 0,06 milímetros (0,0025 de polegada) [Bar13a].

Outra tecnologia de impressão **3D** é a **FDM**, que consiste na utilização de um material semi-líquido (normalmente utiliza-se um polímero termoplástico quente), que é extrudado a partir de um bico com temperatura controlada. Este método é capaz de produzir objetos bastante robustos para um alto grau de precisão. Um dos principais benefícios desta técnica é que os objetos podem ser feitos com termoplásticos tradicionais de moldagem por injeção.

A maioria das impressoras **3D** do tipo **FDM** podem imprimir com o polímero **ABS** como o bioplástico biodegradável **PLA**, que é produzido a partir de óleos alternativos orgânicos. Os desenvolvimentos na área da biologia sintética permitiram a produção direta de **PLA** a partir de diversos materiais.

Além de ser utilizado para produzir objetos poliméricos, as impressoras **FDM** também têm a capacidade de imprimir materiais parcialmente líquidos. As aplicações se diversificaram, sendo capazes de abranger a área de impressão de alimentos até impressoras de concreto. Estas poderão, no futuro, permitir que prédios inteiros (ou grandes partes dos mesmos), sejam impressos em **3D**.

Uma companhia chamada *Objet* desenvolveu um processo chamado *PolyJet Matrix*. Este sistema emite jatos com dois polímeros líquidos fotocuráveis através de uma cabeça de impressão de 96 bicos. Cada camada do objeto é curada por uma luz **UV** após a impressão. Uma das principais vantagens deste processo é a de permitir que a impressão possa utilizar diversos materiais, variando a combinação dos polímeros fotocuráveis lançados pela cabeça de impressão [Obj13].

Outra tecnologia de impressão **3D** existente é a **SLS** ou sinterização seletiva a laser. Esta tecnologia constrói objetos, estabelecendo uma fina camada de pó e, em seguida, utilizando-se um laser para fundir seletivamente alguns de seus grânulos. As impressoras **SLS 3D** podem produzir objetos de diversos materiais em pó. Estes incluem ceras, poliestireno, nylon, vidro, cerâmica, aço inoxidável, titânio, alumínio, ligas de cobalto-cromo etc. Durante a impressão, os grânulos não-ligados (fundidos) servem para apoiar o objeto durante a impressão. Quando este processo for finalizado, quase todo o excesso de matéria-prima poderá ser reutilizado.

Quando a tecnologia **SLS** é utilizada para produzir objetos metálicos, este processo denomina-se Sinterização Seletiva de Metal por Laser (*Direct Metal Laser Sintering*) (**DMLS**). Objetos metálicos originados através de uma impressora **3D DMLS** são cerca de 99,99% densos e, portanto, podem substituir peças de metal na maior parte de suas aplicações [Lay13].

Enquanto o sistema **DMLS 3D** imprime objetos metálicos, é possível utilizar o processo de sinterização a laser para produzir moldes. Um objeto de cera é impresso em **3D**, derrama-se gesso em torno do objeto. Quando aquecida, a cera derrete e é removida, assim o molde de gesso fica pronto para receber metal fundido. Quando o metal e o gesso são resfriados, a peça pode ser retirada e está pronta para sua utilização.

Outra técnica de impressão é a Fusão Seletiva a Laser (*Selective Laser Melting*) (**SLM**). Essa técnica utiliza raio laser para fundir completamente os grânulos de pó que formam um objeto, em vez de apenas aquecer o suficiente para fundi-los juntos. Outra variante, uma técnica chamada Sinterização Seletiva por Calor (*Selective Heat Sintering*) (**SHS**) utiliza uma cabeça de impressão térmica (ao invés de laser), para aquecer as camadas de um pó termoplástico [tC13].

Outra tecnologia de impressão **3D** muito comum é a Multi-jato de Modelagem (*multi-jet modelling*) (**MJM**). Essa tecnologia constrói objetos com sucessivas camadas de pó, com uma cabeça de impressão a jato de tinta, utilizada para pulverizar uma solução ligante que, seletivamente, cola apenas os grânulos que fazem parte do objeto a ser impresso.

Algumas impressoras **MJM** são capazes de pulverizar a solução ligante em quatro cores diferentes, assim, torna-se possível criar objetos **3D** a cores com até $600x540$ dpi de resolução³.

1.4 Condição Atual

Atualmente, o setor de impressão **3D** tem sido englobado por muitas áreas. Para algumas delas são utilizadas impressoras comerciais, tornando-se necessário desembolsar uma quantia superior a 20 mil dólares por unidade [Bar13b]. São impressoras de alta precisão, robustas e de grandes dimensões.

Para impressões esporádicas, pode-se cogitar utilizar um serviço de *bureau online*, com empresas que imprimem objetos, desta forma ampliou-se a utilização da impressão **3D**. Temos como exemplo a *Garagem Fab Lab* que realiza este tipo de serviço.

Para desenvolvedores e interessados no ramo, existem kits à venda. Com eles é possível aprender a construir uma impressora **3D**. Para quem não tem interesse em montar uma máquina, existem empresas, tais como a *Metamáquina*, que vendem impressoras **3D** de baixo custo.

Os objetos impressos normalmente não são utilizados para utilização final, portanto, esta tecnologia era primordialmente considerada para a prototipagem rápida ou para elaboração de moldes.

Na engenharia, a impressão **3D** é utilizada para a realização de testes de produtos antes de sua produção em larga escala, tornando diversas alterações mais rápidas, eficazes e econômicas [Bar13b].

Os ramos de atuação continuam em crescimento: automóveis, joias, brinquedos, máquinas, embalagens, ramos médicos de próteses, órgãos para transplante, recuperação de tecidos vivos, criação de moldes auriculares, criação de coroas, pontes e artifícios dentários, armas de uso livre e de difícil detecção etc [Bar13b].

Alguns setores utilizam a impressão **3D** como produto final, para criar objetos de iluminação, móveis, bandejas, bolsas, joias, óculos, tênis, partes de automóveis, dentre outros [Bar13b].

Futuramente acredita-se na possibilidade de transplante de diversos órgãos utilizando a técnica de impressão **3D**; no aperfeiçoamento e no baixo custo de impressoras para uso pessoal; na melhoria

³Pode-se assistir este processo no vídeo apresentado no sitio <http://www.youtube.com/watch?v=7QP73uTJApx&NR=1>.

da tecnologia de impressão e criação de diversos materiais novos imprimíveis.

Provavelmente, a precisão será aprimorada, obtendo-se peças mais detalhadas, de maior qualidade. Também poderá ser aperfeiçoada a utilização de múltiplas cores na impressão 3D [Bar13b].

Não se sabe ao certo até quanto será possível expandir esta área, mas, por enquanto, esta tecnologia tem sido cada vez mais inovadora e se tornará mais presente no dia-a-dia.

1.5 Objetivos

Os objetivos deste relatório técnico são os de aprimorar o conhecimento na área de impressão 3D.

Serão apresentados projetos de melhorias de fabricação do modelo da *Metamáquina 2*, consistindo no projeto de duas réguas de calibração e do teste de barras.

Será apresentado o projeto e construção de um protótipo de um porta rolo de filamento para a impressora 3D.

Será apresentado o projeto de uma nova impressora 3D com extrusão dupla.

E, por fim, serão apresentados os resultados de diversos testes realizados em bancadas: medição de temperatura, avaliação de vibração durante a operação, extrusão e retração durante a operação e impressão de diversos filamentos de cores, materiais e diâmetros distintos.

1.6 Contribuições

As principais contribuições deste trabalho são as de apresentar resultados de testes e projetos para incrementar o ramo de impressão 3D de baixo custo, nos seguintes aspectos:

- Melhorias funcionais para impressoras 3D;
- Melhorias para a fabricação de impressoras 3D;
- Testes realizados em bancadas a fim de avaliar a qualidade de matérias-primas e para a resolução de determinados problemas;
- Projeto de uma nova impressora 3D com extrusão dupla.

1.7 Organização do Trabalho

No Capítulo 2, serão apresentados os projetos referentes às réguas de calibração, ao teste de barras rosadas, ao porta rolo de filamento e à impressora 3D com extrusão dupla.

No Capítulo 3, serão apresentados os testes referentes às bancadas de medição de temperatura, vibração, extrusão, retração e impressão de filamentos diversos.

Finalmente, no Capítulo 4 são discutidas as conclusões obtidas neste relatório.

O *script* efetuado para a elaboração dos desenhos no *software OpensCAD* encontra-se disponível no Apêndice A.

O *script* efetuado para a elaboração dos testes no *software Arduino IDE* encontra-se disponível no Apêndice B.

Capítulo 2

Projetos Realizados

Neste capítulo serão apresentados os projetos e protótipos realizados pela autora durante a sua atuação na *Metamáquina*. A documentação relacionada é apresentada nos apêndices A e B.

2.1 Régulas de Calibração

A impressora 3D possui barras lisas que são posicionadas, utilizando-se uma peça plástica denominada *bar clamp*, em barras rosadas. Estas barras são apresentadas na cor azul na figura 2.1.

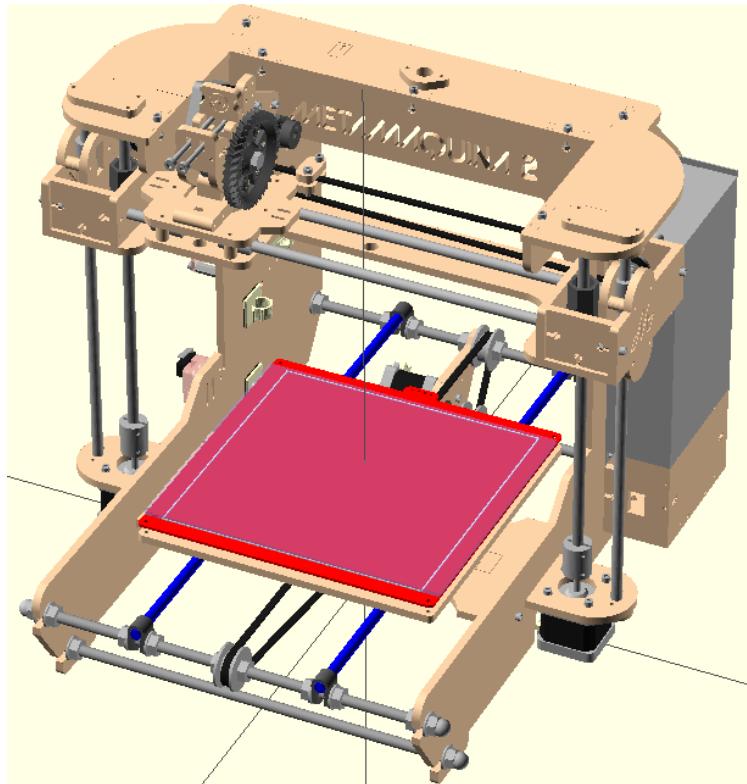


Figura 2.1: Barras lisas a serem posicionadas com precisão.

O processo de calibração destas barras é realizado utilizando-se um paquímetro e rosqueando as porcas adjacentes ao *bar-clamp* até ficar prender com a medida exata. Com a utilização das réguas de calibração este processo se torna muito mais rápido e preciso.

O modelo das réguas e seu posicionamento na impressora 3D é apresentado nas figuras 2.2 (vista frontal) e 2.3 (vista dorsal).

A figura 2.4 são apresentadas as faces para o corte a *laser* as réguas de calibração.

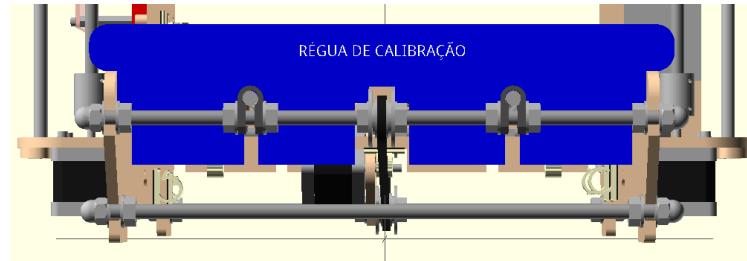


Figura 2.2: Régua de calibração frontal.

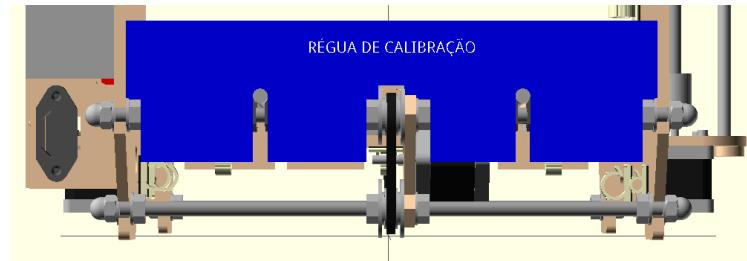


Figura 2.3: Régua de calibração dorsal.

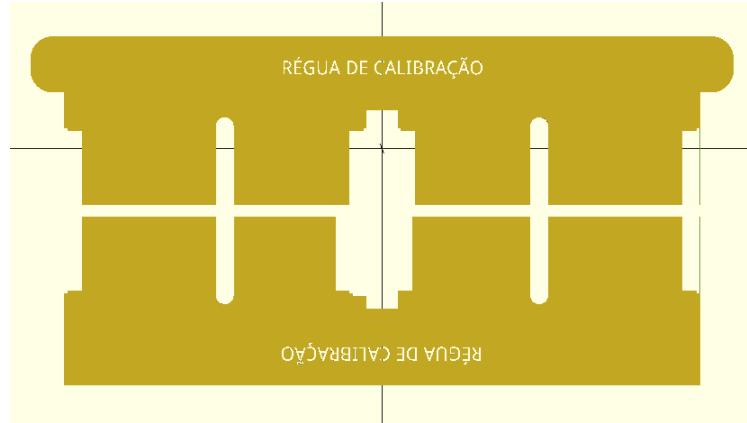


Figura 2.4: Faces para o corte a laser das réguas de calibração.

Infelizmente, estas réguas de calibração não chegaram a ser testadas, já que não foi feito um corte a laser para a elaboração do protótipo.

2.2 Teste de Barras

Foi elaborado um projeto para testar as barras rosadas do eixo Z, a fim de avaliar seus comprimentos e seus graus de empenamento para rejeitar ou aceitar antes da instalação.

Foi elaborado um desenho do modelo em *3D* para o teste, apresentado na figura 2.5 e seu *script* é apresentado no apêndice A.2.

Foram elaborados dois *scripts* para utilizar um botão acionador, a fim de que um motor rotacione a barra. Um *script* corresponde ao movimento de rotação de forma constante e o outro rotaciona de forma gradual para um lado até parar e faz o mesmo no sentido oposto, continuamente. Estes *scripts* podem ser visualizados no apêndice B.1.1 e no B.1.2.

Não foi elaborado um protótipo para este caso, pois o modelo deveria ser aprimorado para ser funcional e poder alocar a fonte e a placa a fim de movimentar o motor e, também, devido ao fato da placa disponível (*Megatronics v1.0*), apresentar problemas para atuar de forma independente do



Figura 2.5: Modelo para o equipamento de teste de barras.

uso do computador.

2.3 Porta Rolo de Filamento

A impressora 3D produzida pela *Metamáquina* possui tecnologia de deposição de polímero através de um bico extrusor. Este polímero é comercializado na forma de rolos de filamento. No mercado existem em dois diâmetros de filamento para impressão 3D, sendo de 1,75mm ou 3,00mm. A *Metamáquina* produz impressoras para filamentos de 3,00mm.

As impressoras da *Metamáquina* podem utilizar como matéria-prima dois polímeros distintos, o **ABS** e o **PLA**, ambos vêm em rolos de mesmas dimensões, portanto, a ideia foi fazer um porta rolo de filamento.

A função deste porta rolo de filamento é o de organizar a área da máquina e impedir a presença de nós e travamentos do filamentos, prejudicando a impressão.

Este porta rolo de filamento foi elaborado utilizando o mesmo padrão da *Metamáquina*, o material é Madeira de Média Densidade (*Medium-density fibreboard*) (**MDF**), utilizando-se tecnologia de corte a laser e com barra rosada e porca de finalização tipo domo.

Na figura 2.6 é apresentado o modelo visual do porta-rolo e na figura 2.7 apresenta-se a face elaborada para possibilitar o corte a laser.

Na figura 2.8 apresenta-se o protótipo do porta rolo de filamento.

Este porta-rolo não foi implementado como produto final, pois, por questões de praticidade de operação e pelo diferente tamanho de rolos de filamento presentes no mercado, é possível torná-lo mais simples e integrado na impressora.

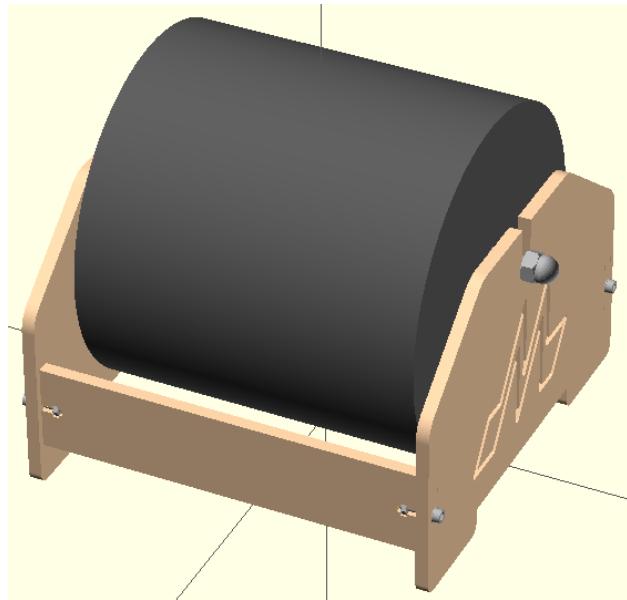


Figura 2.6: Modelo elaborado para o porta rolo de filamento.

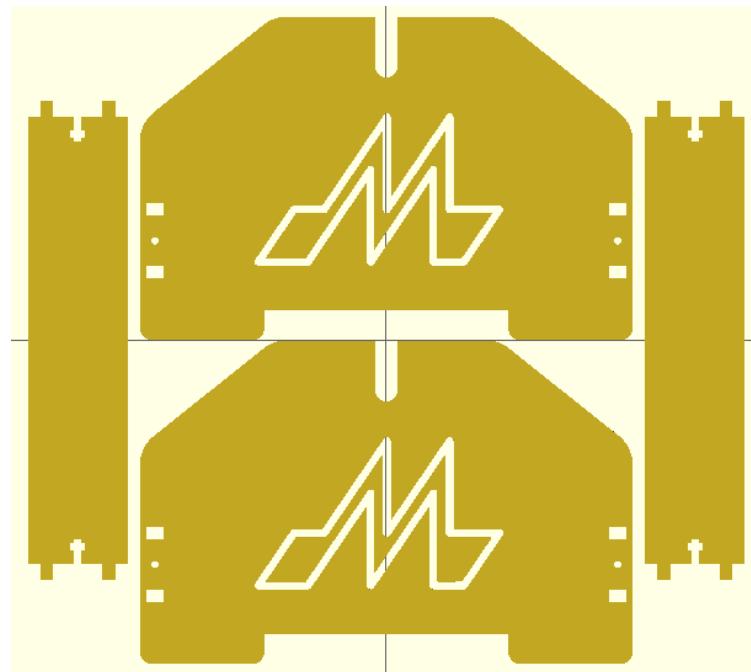


Figura 2.7: Faces para o corte a laser do porta rolo de filamento.

2.4 Impressora 3D com extrusão dupla

Para elaborar o projeto da impressora 3D com extrusão dupla foi aproveitado o projeto da *Metamáquina 2* atual, minimizando-se as alterações a serem implementadas.

A seguir serão apresentados os projetos de *hardware* e *software* da impressora 3D e alguns de seus resultados finais de peças de teste.



Figura 2.8: Protótipo elaborado para o porta rolo de filamento.

2.4.1 Hardware

Para a elaboração desta impressora, foram efetuadas algumas alterações em relação ao modelo da *Metamáquina 2*, que serão descritas a seguir.

Foram instalados dois *kits* da empresa *seemeCNC*, que continham dois extrusores com tubo *Bowden*, dois engates rápidos, dois bicos extrusores de 0,35mm e suportes para prender o extrusor e o motor na carcaça da impressora.

O tubo *Bowden* consiste em um duto de Politetrafluoretileno (**PTFE**) translúcido que transmite o torque do motor ao filamento e permite a sua extrusão na saída do bico. Esse mecanismo permite utilizar um mecanismo de extrusão mais leve e ágil.

Para inserir os *kits* na *Metamáquina* foram necessários efetuar quatro furos na carcaça, visto nas figuras 2.9 e 2.10, e elaborar um prendedor para os dois extrusores no carro X. Esse prendedor foi modelado e construído em plástico **ABS**.

A primeira versão do prendedor é apresentada na figura 2.11 e a versão final é apresentada nas figuras 2.12 e 2.13.

O protótipo do prendedor dos bicos extrusores montado no carro X é apresentado na figura 2.14.

O *script* do modelo do prendedor dos dois bicos extrusores é apresentado no apêndice A.4.

A impressora final pode ser vista na figura 2.15.

2.4.2 Software

A seguir serão apresentados os ajustes efetuados no *firmware*, *slic3r* e no *software Printrun* para uma impressão com extrusão dupla.

Firmware

Para elaborar o *firmware* da impressora, foi utilizado o *software Arduino IDE*.

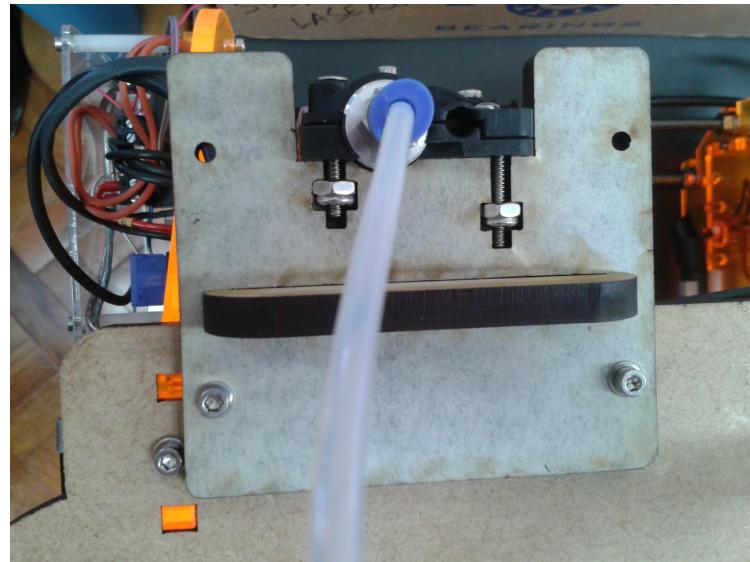


Figura 2.9: Kit de um dos extrusores preso na carcaça da impressora 3D.

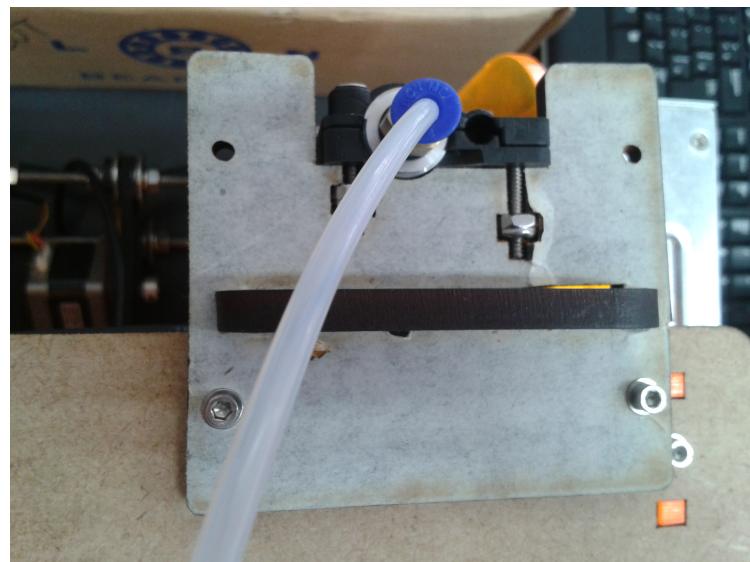


Figura 2.10: Kit de um dos extrusores preso na carcaça da impressora 3D.

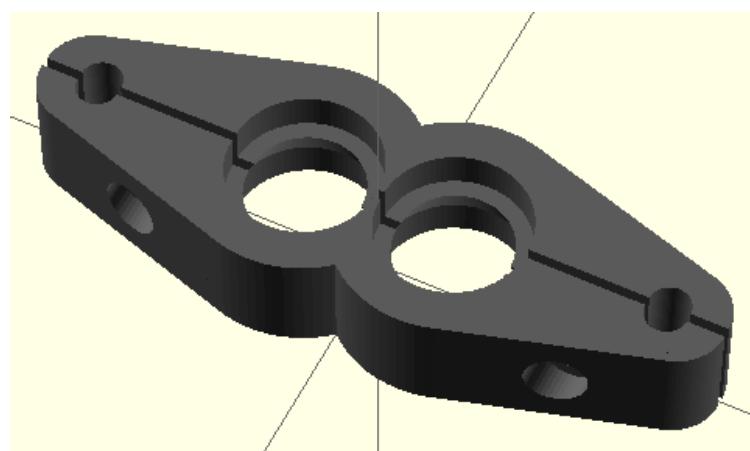


Figura 2.11: Primeira versão do prendedor dos bicos extrusores.

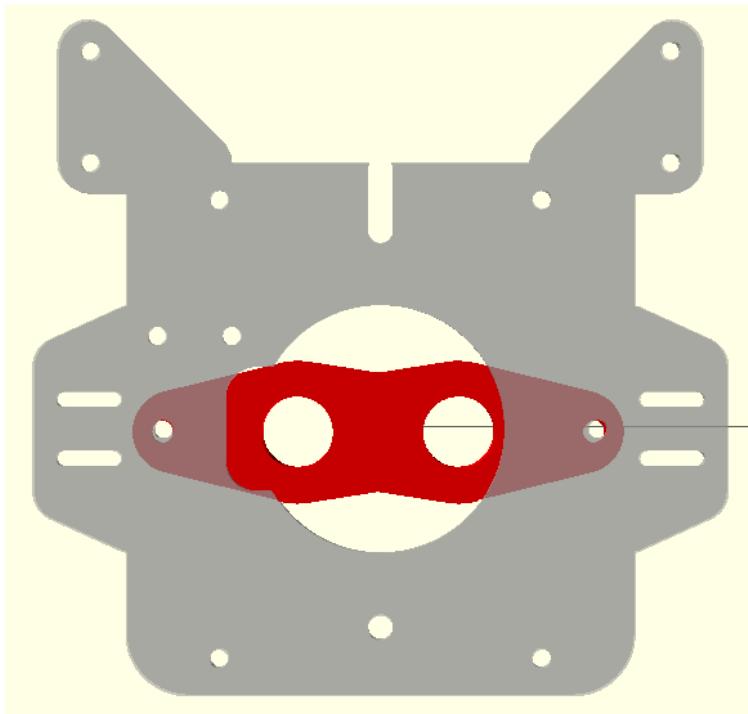


Figura 2.12: Versão final do prendedor dos bicos extrusores - vista superior.

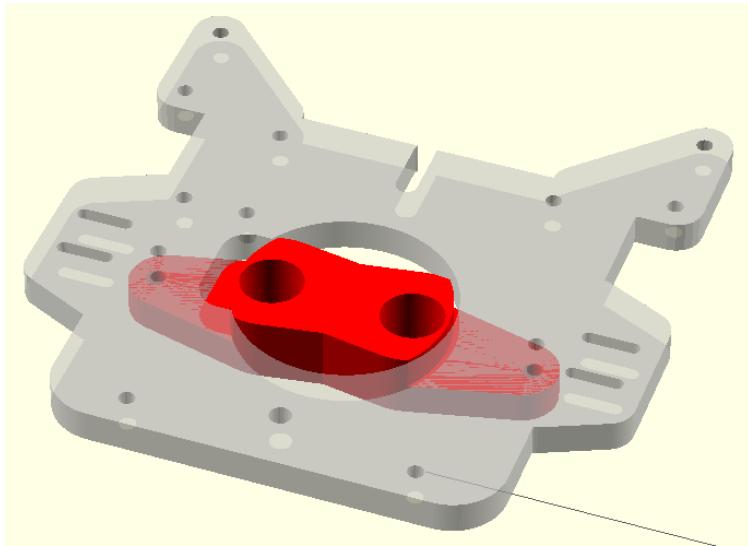


Figura 2.13: Versão final do modelo do prendedor dos bicos extrusores - visão 3D.

O *Arduino* é uma plataforma *open-source*, é também denominado um *software livre*, sua programação é inspirada em código *C/C++*. Esse *hardware* foi apresentado em 2005 com intenção de auxiliar projetos escolares, de forma a ter um orçamento menor que outros sistemas de prototipagem disponíveis naquela época.

É um sistema versátil para fazer projetos, é possível utilizar sensores e motores.

No *Arduino* existem módulos expansivos chamados de *Shields* que servem para estender a sua capacidade. Existe vários tipos de *Shields* (*wifi*, *Bluetooth*, motor, *ethernet* etc).

O *software* do *Arduino IDE* é uma aplicação multiplataforma escrita em *Java*, na qual é derivada dos projetos *Processing* e *Wiring*. Para a estrutura de um arquivo é necessário declarar duas funções:

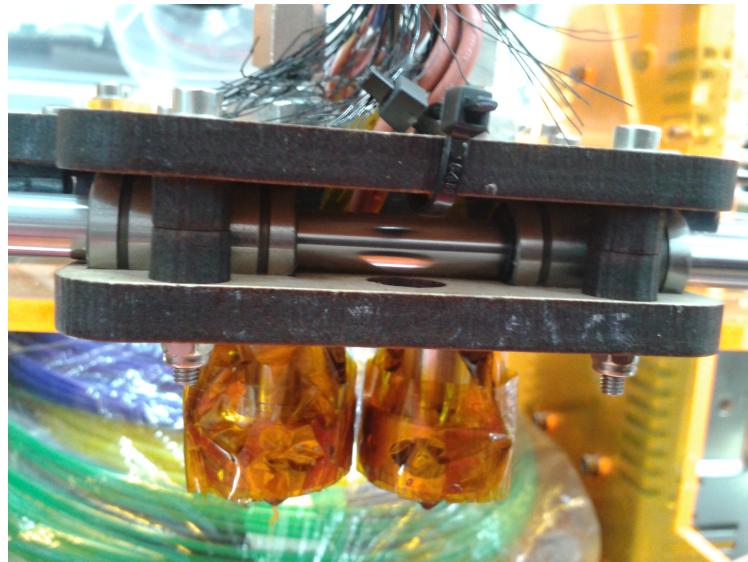


Figura 2.14: Protótipo do prendedor dos bicos extrusores.

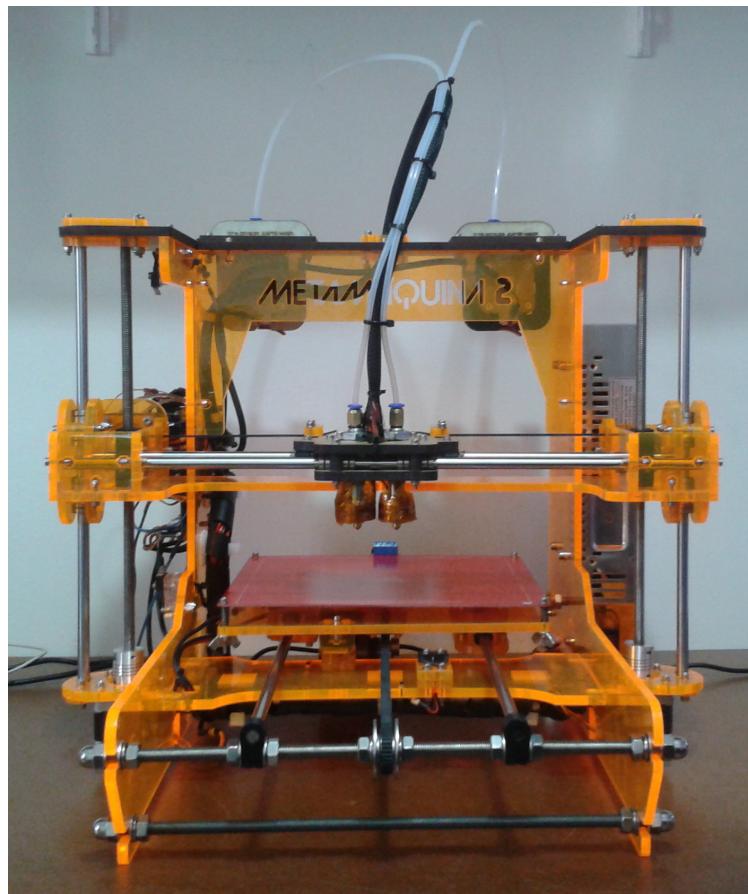


Figura 2.15: Protótipo da impressora 3D.

- *setup()* - Inserida no inicio, na qual pode ser usada para inicializar configuração;
- *loop()* - Chamada para repetir um bloco de comandos ou esperar até que seja desligada.

Para o caso da elaboração da impressora 3D com extrusão dupla, foi utilizado o como base o *firmware* do modelo atual da *Metamáquina* e foi compilado utilizando-se o *software Arduino IDE*, cuja interface é apresentada na figura 2.16.

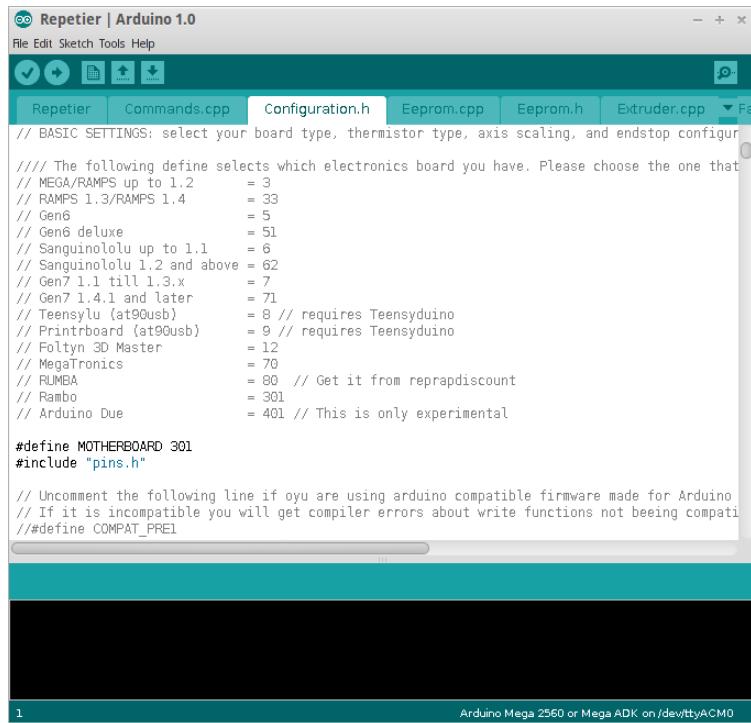


Figura 2.16: Interface do software Arduíno.

As alterações foram feitas no arquivo *configuration.h* do *firmware Repetier*, que é apresentado no apêndice B.2.

Basicamente, o que foi alterado consiste no número de extrusores e no de passos por milímetro, devido ao fato do extrusor da empresa *seemeCNC* possuir sistema *direct drive*, ou seja, inexiste o sistema de redução entre engrenagens, o parafuso trator está acoplado diretamente no eixo do motor.

Outra modificação importante é o filamento utilizado pela impressora 3D ser mais fino (1,75mm).

A impressora com extrusão dupla necessita de uma calibração específica para que os dois extrusores possuam o mesmo sistema de coordenadas e o mesmo ponto inicial. Um ponto importante a mencionar é a altura, os dois bicos devem estar na mesma altura para impedir que um deles danifique as peças impressas.

Para o ajuste da altura, inicialmente regulou-se rosqueando a porca que prende os extrusores, após utilizou-se o alinhamento do eixo Z, pelos acoplamentos presentes nas barras rosadas, para o ajuste de altura dos bicos extrusores.

A seguir, foi feito um ajuste a partir do *software* da impressora, de acordo com a variável *offset* apresentada no arquivo de configuração do *firmware*, vide apêndice B.2, que ajusta a diferença de coordenada dos eixos X e Y entre os dois extrusores, configurou-se um dos extrusores na origem (0,0) e o segundo, quando estava na origem, sofria o desvio configurado pelo *offset*.

Slic3r

Para transformar desenhos tridimensionais no formato de leitura da impressora 3D utiliza-se o *software slic3r*¹.

¹Este *software* pode ser localizado no sítio <http://slic3r.org/>.

Na *Metamáquina 2* o *slic3r* é integrado ao *software* de controle da impressora, e pode ser encontrado no sítio <https://github.com/Metamaquina/Printrun> na pasta *Slic3r_gnulinux_32* ou *Slic3r_gnulinux_64* (dependendo do número de *bits* do computador).

O **STL** é o formato utilizado pelo *Slic3r* para fatiar sólidos modelados, ele não possui dados de cor, textura ou material.

Ao carregar um arquivo do tipo **STL** o *software Printrun* o transforma em um formato específico para impressoras **3D**, o *gcode*, a partir de dados do tipo de filamento(**ABS** ou **PLA**), densidade de recheio e se é necessário criar um suporte para elaborar a peça desejada. Esta janela é apresentada na figura [2.17](#).



Figura 2.17: Interface do Conversão de arquivos **STL** para **.gcode** pelo *Pronterface*.

No *gcode* o desenho é descrito através de códigos de comandos de temperatura, posição, velocidade. Similares ao de um equipamento Controle Numérico Computadorizado (*Computer Numeric Control*) (**CNC**).

O *slic3r* deve sofrer pequenos ajustes para imprimir com dois filamentos, tais como incluir o segundo extrusor, ajustar o tamanho do orifício, determinar a temperatura a ser utilizada durante a primeira camada e o restante da impressão, controlar a vazão de plástico extrudado *extruder multiplier* (consiste em uma porcentagem em relação ao valor definido no *firmware*), dentre outros ajustes.

Tomou-se por base o padrão da *Metamáquina 2*, excetuando-se o valor de temperatura de extrusão, que foi largamente testado, este sofre variações de acordo com as características de material, cor e transparência de cada filamento.

Outro ajuste consistiu na alteração do diâmetro do filamento, de *3,00mm* para *1,75mm*.

Embora não tenha sido utilizado, é possível ajustar a distância entre extrusores pelo *slic3r* e não apenas pelo *firmware* da impressora.

Inicialmente não foram utilizados os ajustes do item *extruder multiplier* do *slic3r*, porém, posteriormente foi averiguada a sua importância para imprimir com filamentos de maior fluidez, tais como os amarelos e alguns translúcidos. Isto ocorre pois alguns filamentos fluem mais que outros na temperatura de impressão, e prejudicam a qualidade das peças produzidas.

Suas telas são apresentadas nas figuras 2.18, 2.19, 2.20 e 2.21.

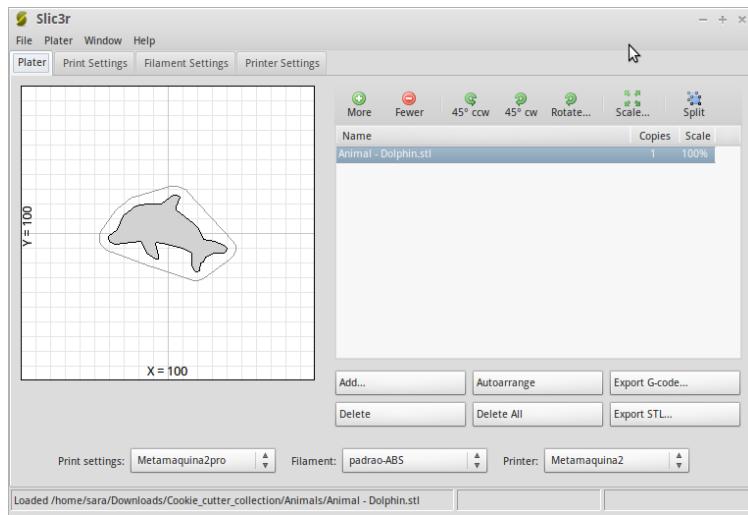


Figura 2.18: Tela geral do Slic3r.

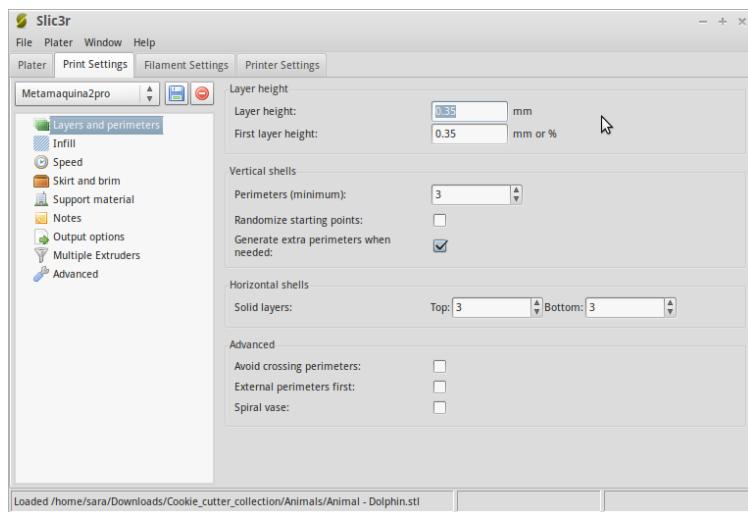


Figura 2.19: Tela do Slic3r de propriedades de impressão.

Na criação de um arquivo *gcode* pelo *Slic3r* para uma impressão com um equipamento de dupla extrusão, cada extrusor é tratado como um arquivo **STL** distinto, é feita a combinação dos dois arquivos, sendo o primeiro selecionado o correspondente ao extrusor 0 e o segundo ao extrusor 1.

Os dois **STLs** após serem combinados geram um arquivo *AMF* e, após sua inclusão no *Slic3r* é possível gerar um arquivo *gcode* que engloba os dois extrusores.

Os comandos para o extrusor 0 são descritos com preâmbulo *T0* e, para o extrusor 1, seu correspondente *T1*. Um trecho de um arquivo *gcode* é apresentado abaixo:

```

1 T0
2 G1 Z0.350 F7800.000
3 G1 F1800.000 E-2.00000
4 :

```

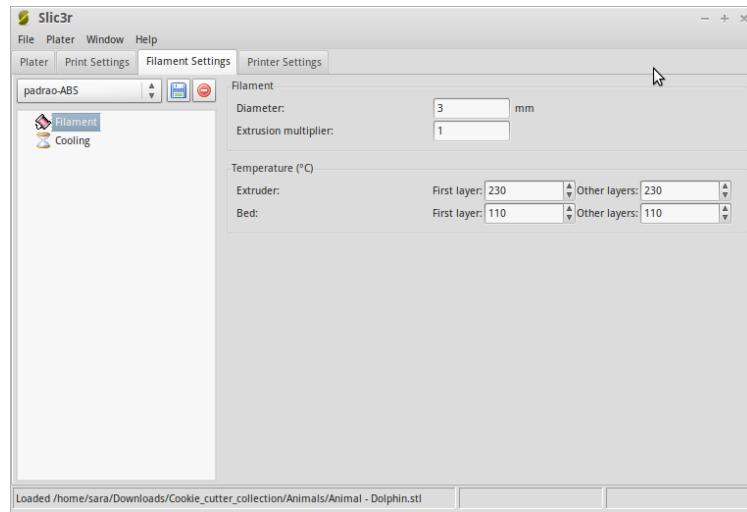


Figura 2.20: Tela do Slic3r sobre propriedades do filamento.

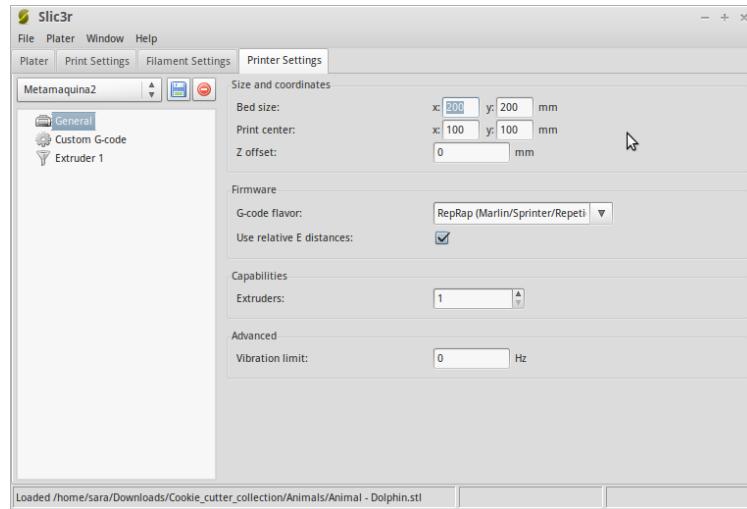


Figura 2.21: Tela do Slic3r sobre propriedades da impressora.

```

5 T1
6 G1 E-2.00000
7 G1 X92.524 Y166.021 F7800.000
8 G1 F1800.000 E2.00000

```

Enfim, o *slic3r* é uma ferramenta poderosa que permite transformar desenhos 3D em formato legível pela impressora e é capaz de trabalhar com extrusão múltipla, sendo um meio eficiente e prático para sua implementação em impressoras 3D.

Pronterface

O *Pronterface* consiste na interface gráfica de acompanhamento e controle da impressora 3D. Para adquirir basta baixá-lo utilizando-se o link <http://metamaquina.com.br/downloads/>. A tela é apresentada na figura 2.22.

É possível passar comandos manuais, realizar a calibração de altura, acompanhar o aquecimento da base e do bico extrusor e visualizar as partes que estão sendo impressas *online* durante todo o tempo de operação do equipamento.

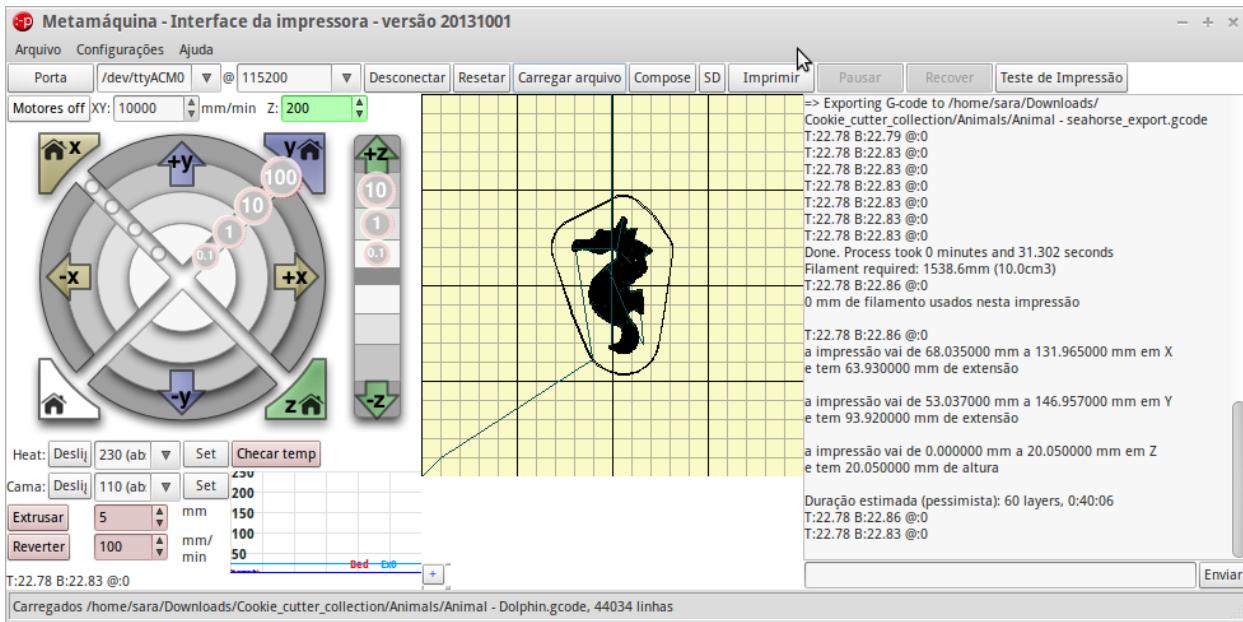


Figura 2.22: Tela do Pronterface.

2.4.3 Impressões Realizadas

Foram impressas peças em duas cores a fim de calibrar a impressora, esta seção apresentará as fotos de peças prontas e os principais problemas encontrados durante a impressão de testes de calibração e peças.

Na figura 2.23 é apresentada a primeira peça impressa, conhecida como o chaveiro de teste da *Metamáquina*², foi impresso com filamento **ABS** na cor branca.



Figura 2.23: Primeira peça impressa.

Após, foram efetuados testes de calibração para dois filamentos, o modelo é encontrado no sítio <http://www.thingiverse.com/thing:124450>. Na figura 2.24 é apresentada uma impressão de duas cores, utilizando filamentos de **ABS** na cor branca e rosa.

²Este modelo pode ser encontrado no sítio <https://github.com/Metamaquina/Printron>

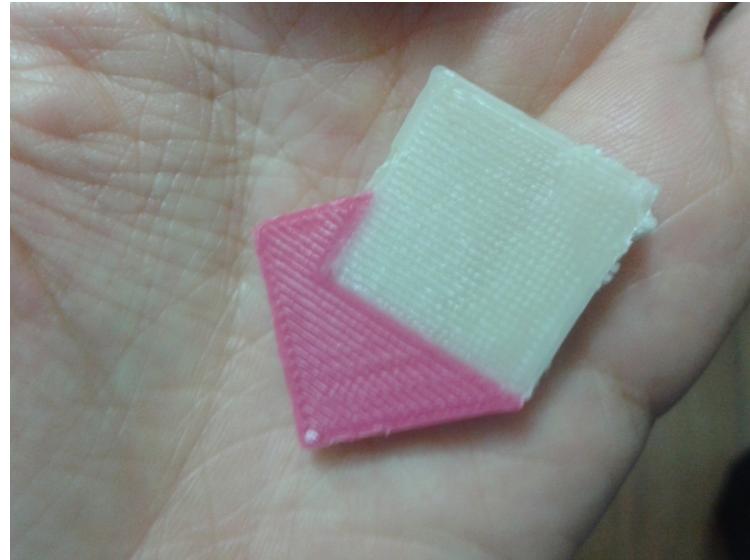


Figura 2.24: Impressão em **ABS** em duas cores.

Por fim, é apresentado o teste de uma peça de duas cores, filamento em **ABS** branco e em **PLA** laranja na figura 2.25.



Figura 2.25: Impressão em **ABS** e **PLA**.

Durante a tentativa de extrusão de peças maiores ocorreram problemas de derretimento do filamento dentro da conexão do tubo *Bowden*. Para resolver este problema é necessário utilizar *coolers*, a fim de refrigerar os engates e impedir a torção do filamento e seu emperramento.

Também ocorreram problemas de derretimento do filamento ao passar pelo parafuso trator, caso de ocorrência rara. Para resolvê-los é necessário utilizar *coolers*, a fim de refrigerar os engates e impedir a torção do filamento e seu emperramento. Outra solução possível é a de diminuir a corrente no motor.

Estas soluções descritas anteriormente ainda não foram testadas, fator necessário para dar continuidade a este produto.

Também é necessário comentar que estes problemas não foram encontrados para todos os filamentos testados, apenas para as cores azul-escuro e prata.

Outros problemas encontrados foram a perda de passo, corrigida com o aumento de temperatura do bico extrusor, e o descolamento da base, corrigida com o aquecimento da base ou utilização de fita de poliamida tipo *Kapton*. Os filamentos translúcidos apresentaram maiores problemas.

Outra solução a fim de aprimorar a qualidade de impressão para os filamentos é a de alterar o fator multiplicativo de extrusão presente no *Slic3r*, que não foi abordado.

Por fim, com este protótipo, tornou-se possível imprimir peças em dois materiais e/ou cores diferentes, abrindo um novo rumo na área de impressão **3D** na *Metamáquina*.

Capítulo 3

Testes Realizados

Foram realizados diversos testes a fim de ter maior controle de qualidade para o material recebido e para solucionar possíveis óbices encontrados com determinadas alterações, tais como: a da composição e fabricante de filamentos, a de termistores e problemas de soltura de porcas e parafusos da carcaça e partes componentes da impressora.

A seguir serão descritos os maiores desafios enfrentados e os experimentos efetuados em bancadas de testes.

3.1 Temperatura

A bancada de testes de termistores surgiu devido ao fato destes serem comprados em lotes, da marca *Addtherm*, e o modelo ter sido alterado.

Embora seja encontrado na folha de dados do fabricante os parâmetros de β e R_0 , a tabela obtida a partir destes dados não se mostraram compatíveis.

Para o primeiro lote, a tabela de temperaturas era conhecida, pois foram levantados os pontos experimentalmente. Entretanto, para outros dois lotes e para os futuros era importante a elaboração de experimentos para a aferição da temperatura medida pelo termistor.

A seguir serão apresentados alguns conceitos básicos sobre o tipo de termistor utilizado pela *Metamáquina* e os testes efetuados, a fim de corrigir a tabela de temperatura dos termistores utilizados.

3.1.1 Conceitos

Termistores consistem em um tipo de resistores que sofrem grandes variações de acordo com a mudança de temperatura. Os resistores utilizados na *Metamáquina* são de vidro e podem ser visualizados na figura 3.1.

O intervalo de temperatura medido pelo termistor é adequado ao utilizado durante a impressão, que oscila entre $150 - 250^{\circ}C$.

Termistores seguem um comportamento de variação da resistência com a temperatura, pode-se observar que o termistor de vidro da *Addtherm* é considerado um Coeficiente Negativo de Temperatura (*Negative Temperature Coefficient*) (NTC), onde uma variação positiva de temperatura causa uma variação negativa na sua resistência.

Duas equações largamente utilizadas para a modelagem destes termistores são as de *Steinhart-Hart* e sua versão simplificada, a do parâmetro β , que é largamente utilizada pelos fabricantes. A equação de *Steinhart-Hart* é apresentada na 3.1.

Os parâmetros de a , b e c são denominados parâmetros de *Steinhart-Hart* devem ser especificados para cada termistor, normalmente fornecidos pelo fabricante ou obtidos através de experimentos

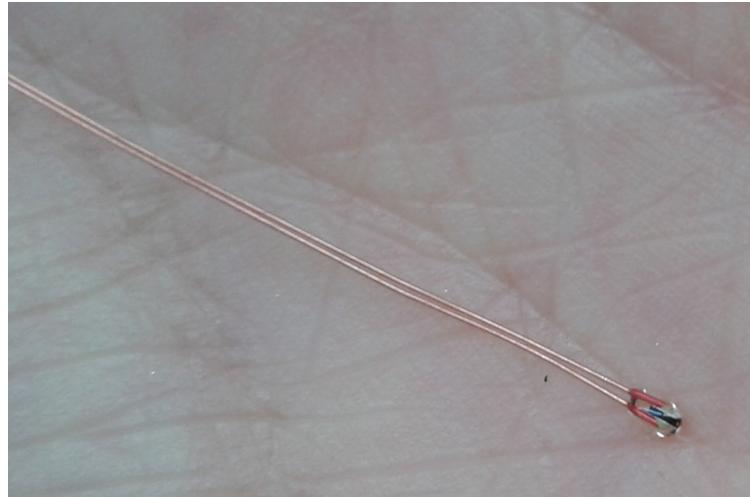


Figura 3.1: Termistores de vidro da addtherm.

[McG88].

$$\frac{1}{T} = a + b \cdot \ln(R) + c \cdot \ln(R)^3 \quad (3.1)$$

A equação do parâmetro β é apresentada na 3.2 [McG88]:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} \cdot \ln \left(\frac{R}{R_0} \right) \quad (3.2)$$

Ao instalar o termistor em uma placa, tal qual a utilizada pela *Metamáquina*, no caso uma *Rambo v1.0*, o valor lido é denominado *RawADC*.

Portanto, o valor lido pela placa é um sinal de corrente, para ser transformado em valor de temperatura ele passa por três transformações. A primeira é devida à transformação do sinal digital para analógico. A segunda é obtida a partir da equação proveniente do diagrama esquemático (figura 3.2), que consiste em um sinal lido com duas resistências em paralelo. A terceira é obtida através da equação do parâmetro β , descrito anteriormente 3.2.

Desta forma, podemos partir para os experimentos realizados, levantar curvas de aferição intra e extra lotes e averiguar a qualidade e a precisão do produto recebido.

3.1.2 Experimentos

Foram realizados testes para os três lotes de termistores, o primeiro já previamente adaptado e em estado funcional e os demais, que apresentaram problemas e tiveram de ser aferidos.

Um termopar tipo J foi utilizado como medidor padrão de temperatura, sendo considerado como o elemento que apresentava a temperatura correta a ser lida. Todos os termistores foram calibrados com este instrumento.

Os testes foram realizados utilizando um termopar e um termistor sendo lido pelo próprio *Printerface* da impressora, o *firmware* da placa *Rambo* permite mensurar a temperatura de até dois termistores concomitantemente. Este teste poderia ser aprimorado, mas inicialmente existiam limitações físicas, devido à proximidade do orifício onde era alocado o termistor e a resistência, ambos devidamente instalados no bico extrusor.

Thermistors

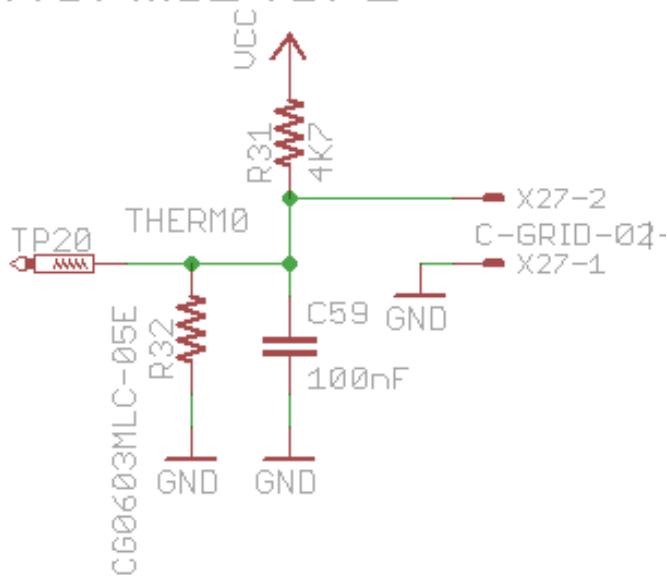


Figura 3.2: Desenho esquemático de um dos termistores da placa Rambo.

A placa *Rambo* possui quatro entradas para termistores[Rus13], ou seja, estes ensaios podem ser expandidos para ser possível a leitura deste número de termistores concomitantemente. Foi elaborado um *script* para o *firmware* deste experimento e pode ser observado no apêndice B.3.

O grupo de termistores testados pode ser observado na figura 3.3, que corresponde a um bico extrusor com múltiplos furos e extrusores instalados.

Estes termistores eram presos em barras metálicas e inseridos em um recipiente, a fim de proteger o experimento de interferências externas, além de proteção pessoal, já que foram testados em altas temperaturas impróprias para o contato humano.

Foi utilizado o controle de temperatura pelo *software Pronterface*, anotado o valor de *RawADC* correspondente ao termistor a ser aferido e o valor de temperatura do termopar.

Com os valores originados pelos experimentos foi possível construir o gráfico 3.4, nota-se a divergência entre os valores medidos para cada lote, este era o principal motivo de leituras errôneas e impressões de má qualidade e até danos à impressora 3D.

Com estes dados foram consideradas três tabelas a serem utilizadas para seu respectivo lote, solucionando a maior parte dos problemas de impressão encontrados.

3.2 Vibração

Outro teste realizado foi o de vibração, necessário devido à ocorrência de soltura de porcas, arruelas e parafusos da carcaça e partes componentes da impressora 3D.

O teste teve 474 horas de duração, e consistiu em movimentar o carro extrusor ao longo do eixo X em movimentos curtos e rápidos, a fim de provocar uma forte vibração por toda a impressora.

Foram elaborados dois *scripts* em linguagem *Python*, apresentados no apêndice B.4. Para este



Figura 3.3: Teste de termistores.

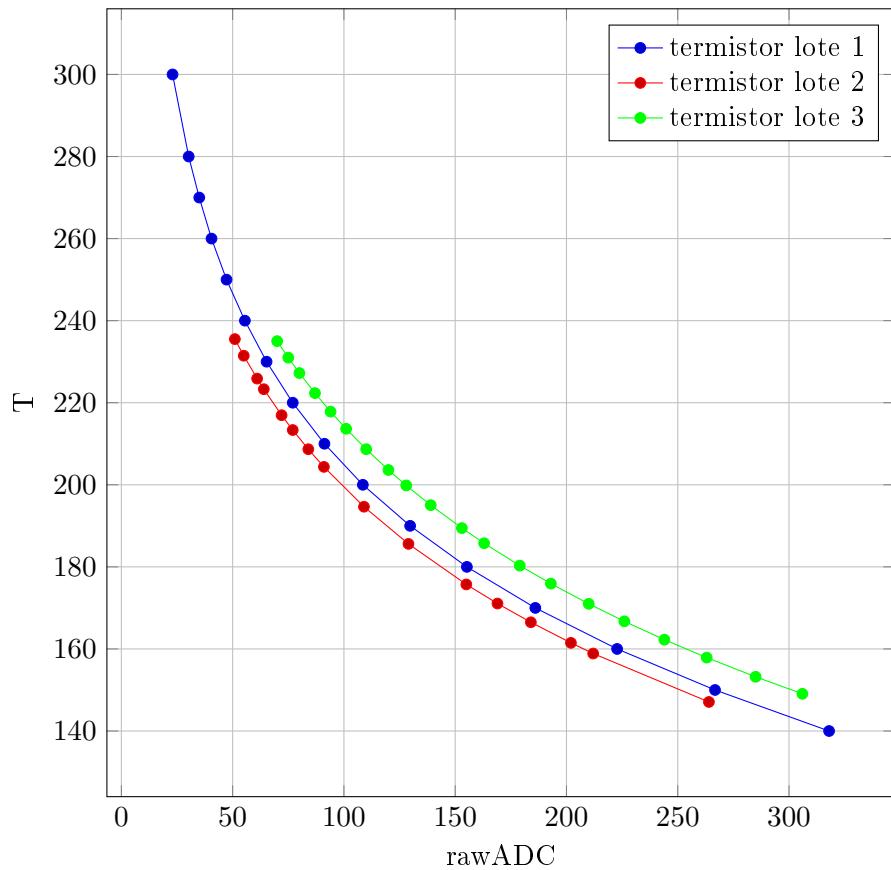


Figura 3.4: Média dos resultados dos ensaios para os três lotes de termistores.

Este teste era importante criar um sistema repetitivo de uma sequência de movimentos. Por este motivo, descartou-se a possibilidade de utilizar o *Printrun*, por e preferiu-se utilizar a ferramenta *Printcore* através do uso do terminal.

Foi elaborado um arquivo *gcode* que determinava a sequência a ser repetida, ou seja, a vibração de ida desde a origem até o fim do curso X e a volta até a origem. Este arquivo é apresentado no apêndice B.4.1.

Para transformar este arquivo para o formato *gcode* é necessário compilá-lo utilizando a seguinte sintaxe via terminal:

```
1 $ cd /[Caminho onde se encontra o arquivo.py]
2 $ python [arquivo.py]
```

O mesmo roteiro deve ser utilizado para compilar outros arquivos em *Python*.

Um outro arquivo utiliza o aplicativo *Printcore* (que está embutido no *Pronterface*, porém não tem interface gráfica), para o controle da impressora, conectando e enviando sinais mandatórios de posição, de acordo com o *gcode* apresentado anteriormente. Este arquivo é ilustrado no apêndice B.4.2. Esta forma permite a geração de comandos e o envio para a impressora infinitamente, este é o motivo de ter sido utilizada.

Foi utilizada uma carcaça de um modelo antigo de impressora 3D da *Metamáquina* e instalou-se o carro com o sistema de extrusão de filamento, que não seria funcional neste caso. Consideraram-se as peças deste carro como as que mais se desprendiam. Assim, nestes pontos críticos, foi utilizada cola trava-rosca para fins de teste.

A máquina utilizada para o teste de vibração é apresentada na figura 3.5. As regiões com utilização de cola trava-rosca são apresentadas na figura 3.6.



Figura 3.5: Impressora Utilizada para o Teste de Vibração.

As figuras 3.7, 3.8, 3.8, apresentam o resultado final do teste dos elementos com cola da marca *Loctite*, de número 243. A figura 3.9 apresenta o resultado utilizando-se cola de alto torque, da marca *Tek Bond* número 128 de alta resistência, aplicada nos parafusos do motor do extrusor.

Para fins comparativos, em um dos parafusos foi utilizada uma porca parloque (auto-travante) e em outro parafuso foi instalada uma porca comum, ambos são apresentados nas figuras 3.10 e 3.11. Nenhuma das porcas se soltaram.

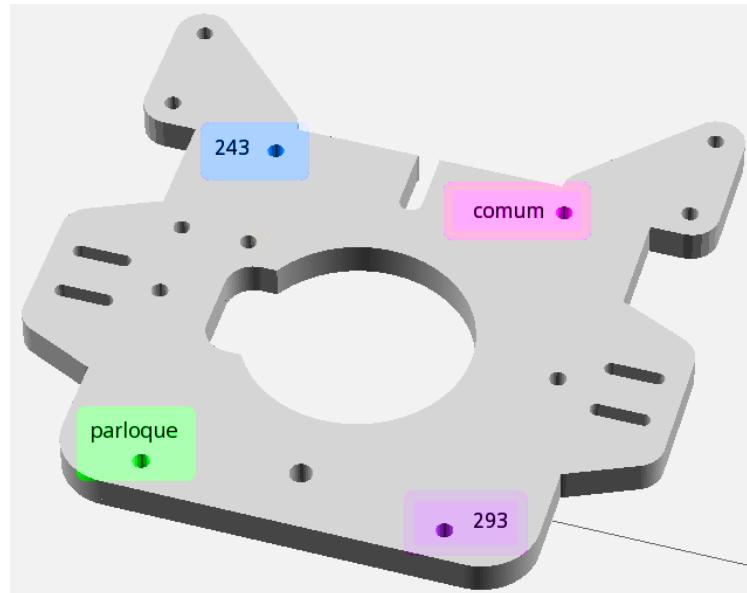


Figura 3.6: Disposição dos principais elementos testados.

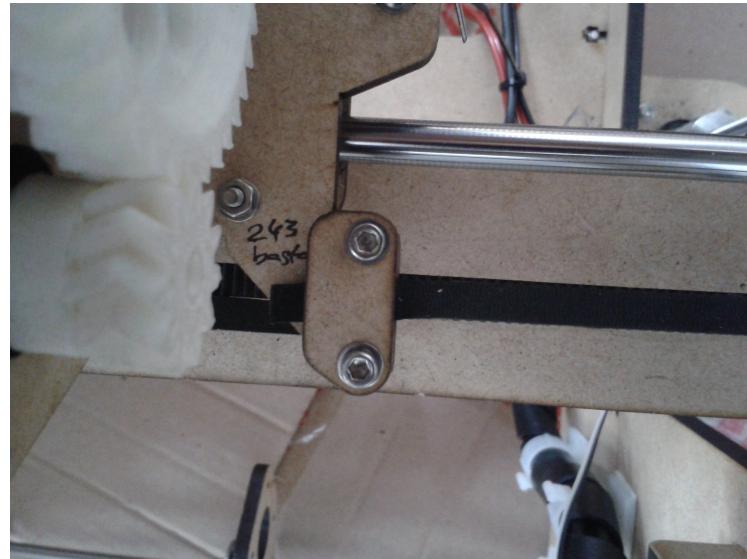


Figura 3.7: Parafuso com porca com grande quantidade de cola trava-rosca Loctite 243.

A cola trava-rosca *Loctite 243* é de alto torque e não permite a desmontagem caso seja necessário. Os dados técnicos desta cola podem ser encontrados na referência [Loc13].

Embora as regiões monitoradas não tenham se soltado durante a realização do teste, ocorreram algumas avarias importantes a serem destacadas, tais como a queda de parafusos da carcaça e o afrouxamento da correia do carro extrusor.

As regiões onde foram aplicadas as colas não se soltaram, esta solução mostrou ser eficaz e passível de ser aplicada, principalmente nas regiões onde os parafusos se soltam mais facilmente ou necessitam de uma colocação com folga.

Outra solução, mais aplicada para a carcaça e peças que utilizam encaixes de **MDF**, com porcas foi utilizada pela empresa *seemeCNC* e mostrou-se interessante e, portanto, também sugerida. Trata-se de alterar a dimensão destes encaixes (denominados *T-slots*) para permitir a entrada da porca na sua região sextavada mais estreita, impedindo a sua rotação e facilitando a sua montagem.



Figura 3.8: Parafuso com porca com pouca quantidade de cola trava-rosca Loctite 243.



Figura 3.9: Parafusos com porca com cola trava-rosca Tek Bond 128 - identificados em vermelho.

Vale lembrar que inicialmente a *Metamáquina 2* era montada com porcas comuns, após a verificação da soltura de partes foram implementadas as porcas auto travantes e após a cola. A porca auto travante é de difícil montagem, tornando o processo moroso. A cola auto travante também



Figura 3.10: Parafuso com porca comum.



Figura 3.11: Parafuso com porca comum.

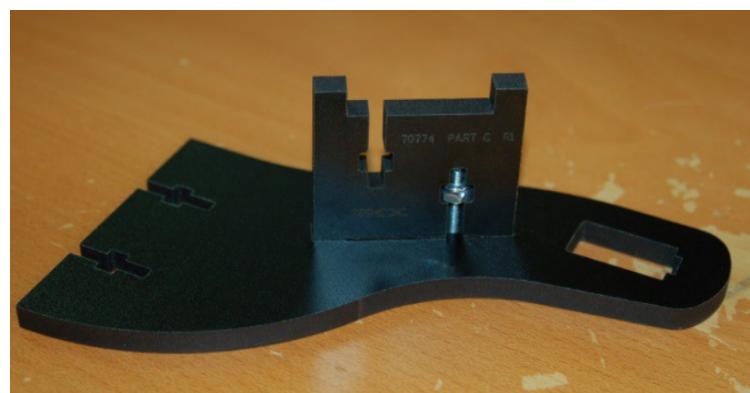


Figura 3.12: T-slot justo para a porca auto travante da empresa seemeCNC [[see13](#)].

apresenta desvantagens, tais como apresentar forte odor, saber aplicar e dosar a quantidade correta.

Outro teste que poderia ter sido implementado para melhor avaliar a soltura de porcas é o

transporte da impressora 3D em largas distâncias, já que notou-se a soltura antes de chegar ao cliente.

3.3 Extrusão e Retração

A ideia da elaboração do teste de extrusão e retração surgiu devido a problemas de superaquecimento e derretimento do filamento durante sua passagem pelo parafuso trator. Este teste consistiu na elaboração de uma sequência de extrusões e retrações de forma contínua, com duração de algumas horas.

O Pronterface foi carregado com um arquivo, de formato *gcode*, elaborado a partir de um arquivo com formato *Python*, apresentado no apêndice B.5.

Para compilá-lo utilizou-se a mesma sintaxe via terminal do teste de vibrações:

```
1 $ cd /[Caminho onde se encontra o arquivo.py]  
2 $ python [arquivo.py]
```

Foi utilizada uma carcaça de uma impressora 3D com extrusor fixo para fazer o teste, implementou-se uma sequência variável de extrusões e retrações, simulando o comportamento de uma impressão complexa.

Notou-se que, inicialmente, durante a extrusão, o filamento parava de sair e era danificado pelo parafuso trator, repetindo o óbice enfrentado durante a operação das máquinas. Porém, ao se ajustar a temperatura, de acordo com os valores obtidos através dos testes de temperatura 3.1 este desafio foi resolvido.

Também foi testada a extrusão utilizando-se tubo *Bowden*, verificado nas imagens 3.13 e 3.14. Estes ensaios não foram conclusivos devido ao fato do material do tubo utilizado (polietileno), ser inadequado para esta finalidade (o correto é de PTFE).

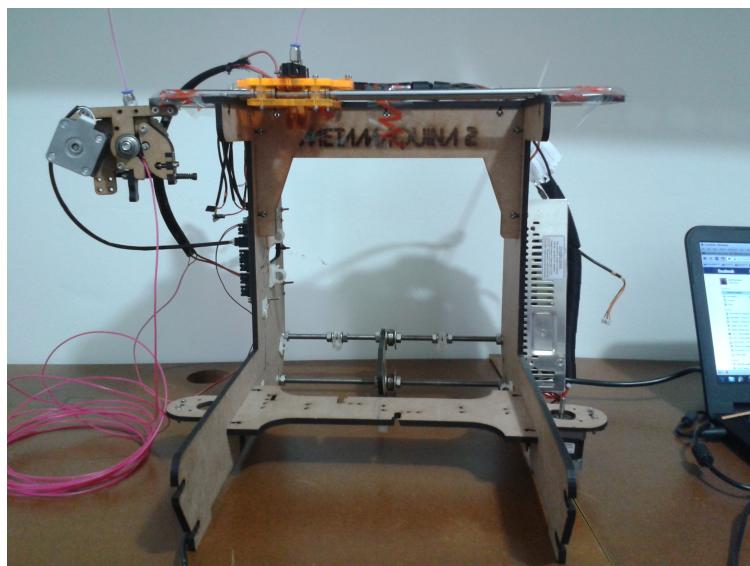


Figura 3.13: Equipamento utilizado para o teste de extrusão adaptado para uso de tubo Bowden.

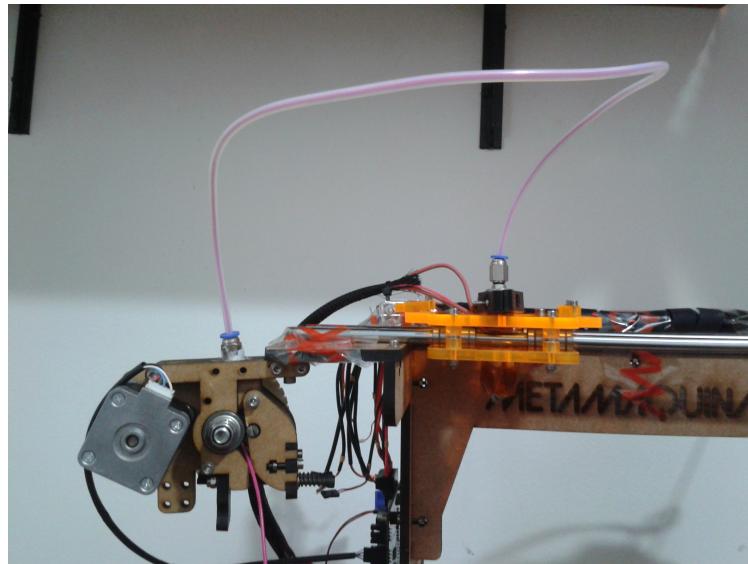


Figura 3.14: Detalhe do ensaio de extração com tubo Bowden.

3.4 Teste de Filamentos

Ao montar os *kits* dos extrusores da empresa *seemeCNC*, notou-se que na parte interna do extrusor havia um tubo de **PTFE** cortado transversalmente.

Assim, surgiu a ideia de flexibilizar a *Metamáquina 2*, que utiliza filamento de 3,00mm para que, numa simples inserção de um tubo de **PTFE** cortado transversalmente, seja capaz de imprimir com filamentos de 1,75mm.

Este procedimento de conversão, que consiste em inserir este tubo, é simples e rápido, além de não ser necessário desmontar o bico extrusor da impressora para reverter este processo.

Inicialmente foi realizada a impressão a partir de um filamento de 3,00mm **PLA** cor amarela apresentada pelas figuras 3.15 e 3.16.

Para realizar a conversão para utilizar filamentos de 1,75mm basta utilizar o tubo apresentado na figura 3.17.

Limpar o extrusor, para eliminar resquícios do filamento de 3,00mm da impressão anterior, conforme apresentado pela figura 3.18.

Inseri-lo no extrusor, conforme apresentado pela figura 3.19 até a altura indicada pela figura 3.20.

A seguir é realizada a impressão com filamento de 1,75mm conforme apresentada pela figura 3.21 e a peça impressa indicada pela figura 3.22.

Este procedimento de conversão foi bem sucedido sendo possível de inclusão no modelo da *Metamáquina 2*.

A seguir serão apresentados os testes realizados em diversos filamentos, a fim de avaliar fornecedores e, também, para obter perfis de impressão compatíveis, já que cada filamento possui uma determinada condição para qualificar uma impressão desejável.

Os testes realizados com filamento **PVA** são complexos, pois devem ser realizados com retração

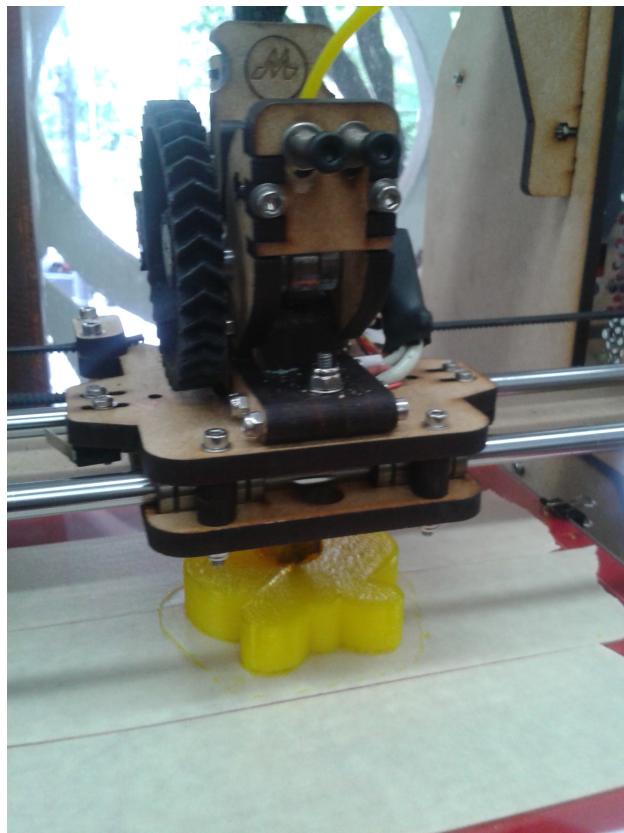


Figura 3.15: Procedimento de impressão para filamento de 3,00mm.



Figura 3.16: Peça final impressa.



Figura 3.17: *Tubo conversor para utilizar filamentos de 1,75mm.*

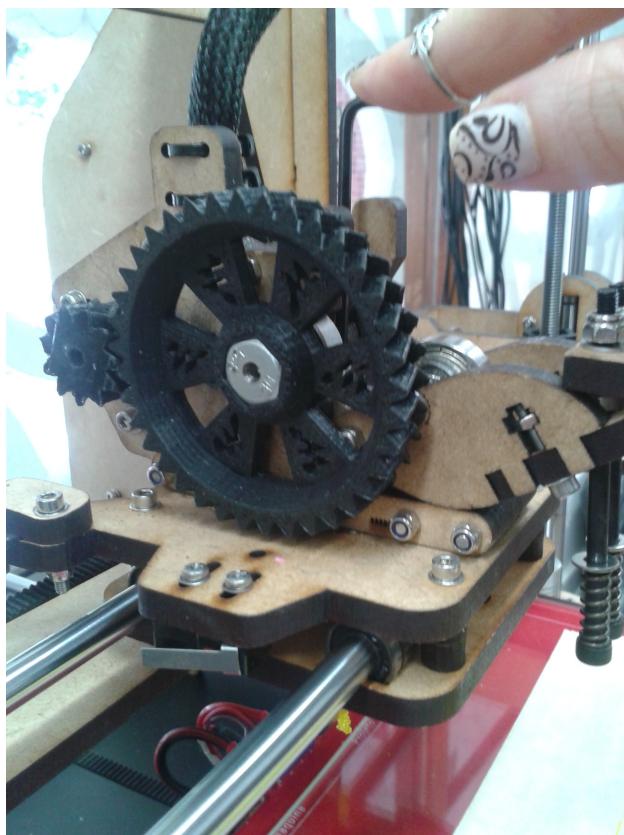


Figura 3.18: *Limpeza do bico extrusor.*

mínima, velocidade baixa e uma guia que impede o enrolamento do filamento dentro do extrusor na região inferior do parafuso trator. Estas guias são apresentadas na figura 3.30 e o seu teste é apresentado na figura 3.31.

O diâmetro dos filamentos pode interferir na impressão, principalmente se ocorrerem imperfeições tais como a variação de diâmetro ao longo do rolo e a ovalização. Os valores mensurados com paquímetro dos diâmetros dos filamentos são apresentados na tabela 3.1. Alguns valores apresentados nesta tabela não estavam disponíveis (Não Disponível (ND)) e também foi mensurado um filamento de material resistente (**HIPS**).

Os filamentos são enumerados conforme o seu Número Identificador (**Id**).



Figura 3.19: Inserção do tubo conversor.



Figura 3.20: Visualização final da colocação do tubo conversor.



Figura 3.21: Procedimento de impressão para filamento de 1,75mm.



Figura 3.22: Peça final impressa.



Figura 3.23: Teste com filamento **PLA** da marca Esun cor azul claro.



Figura 3.24: Teste com filamento **PLA** translúcido da marca Esun cor azul.



Figura 3.25: Teste com filamento **PLA** da marca Esun cor cinza.

Como foram testados diversos filamentos, reuniram-se os parâmetros de fatiamento, inseridos no *slic3r*, em uma tabela, vista na 3.2.

Através dessa tabela nota-se que os filamentos opacos operam com perfis similares aos já pro-

Material	Empresa/Referência	D_{nom}	T_{fab}	Cor	\bar{x}	s	Id
ABS	<i>Esun-China</i>	1,75	220-260	Cinza (<i>Gray</i>)	1,77	0,02	01
				Preto (<i>Black</i>)	1,75	0,02	02
				Azul Claro (<i>Light Blue</i>)	1,74	0,01	03
	<i>Esun-China</i>	1,75	190-220	Azul (<i>Blue</i>)	1,73	0,02	04
	<i>Esun-China</i>	3,00	220-260	Natural	2,98	0,06	05
HIPS	<i>Esun-China</i>	3,00	220-260	Natural	2,95	0,03	06
PLA	<i>Esun-China</i>	3,00	190-220	Branco (<i>White</i>)	2,93	0,05	07
				Amarelo (<i>Yellow</i>)	2,97	0,03	08
				Prata (<i>Silver</i>)	1,77	0,02	09
	<i>ND-China</i>	1,75	ND	Branco Natural (<i>Natural White</i>)	1,78	0,02	10
PVA	<i>Macomp</i>	1,75	ND	Laranja (<i>Orange</i>)	1,86	0,19	11
				Laranja	1,70	0,03	12
				Verde	2,87	0,05	13
				Azul	2,97	0,06	14
				Vermelho	2,50	0,47	15
				Transparente	2,33	0,51	16
PVA	<i>Esun-China</i>	1,75	190-220	Natural	1,73	0,02	17
		3,00			2,88	0,02	18

Tabela 3.1: Verificação do Diâmetro dos Filamentos.

Id	T_{te}	T_{tb}	Ret	$Mult$	Guia	v_{max}
01	230	110	1,0	0,98	não	60
02	230	110	1,0	0,98	não	60
03	220	110	0,5	1,05	não	60
04	220	110	0,5	1,05	não	60
05	230	110	2,0	0,98	não	60
06	220	110	2,0	0,98	não	60
07	230	110	2,0	0,98	não	60
08	230	110	1,0	0,80	não	60
09	230	110	2,0	0,98	não	60
10	230	110	2,0	0,98	não	60
11	210	100	2,0	0,98	não	60
12	200	100	1,0	1,00	não	60
13	200	100	1,0	1,00	não	60
14	200	100	1,0	1,00	não	60
15	200	100	1,0	1,00	não	60
16	200	100	1,0	1,00	não	60
17	195	100	0,2	0,90	sim	40
18	195	100	0,2	0,90	sim	40

Tabela 3.2: Perfis de fatiamento.



Figura 3.26: Teste com filamento **PLA** da marca Esun cor preta.



Figura 3.27: Teste com filamento **PLA** translúcido da marca Esun cor verde.



Figura 3.28: Teste com filamento **HIPS** da marca Esun cor natural.

gramados no *Pronterface* com a alteração da temperatura de extrusão, pois os filamentos da marca *Esun* possuem temperaturas mais altas.

Os filamentos translúcidos operam com retrações menores e com o parâmetro de multiplicador de vazão de extrusão inferior aos dos polímeros opacos.



Figura 3.29: Teste com filamentos **PLA** da marca Macomp.

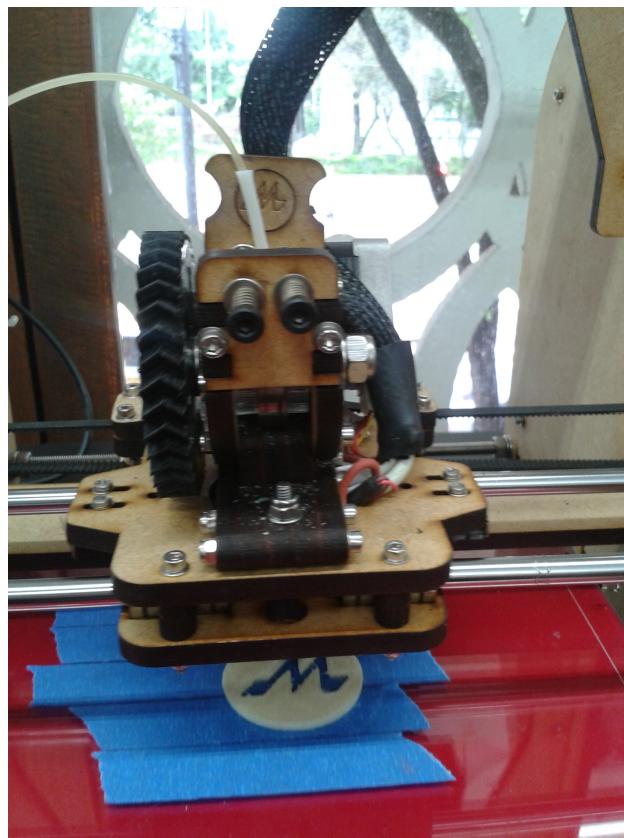


Figura 3.30: Impressão com filamento **PVA** da marca Esun.

O material **PVA** possui maior dificuldade para se obter uma impressão de qualidade, por ser muito flexível é necessário utilizar guias, velocidade de extrusão baixa, retração pequena em relação aos perfis de impressão de outros materiais citados anteriormente no *Pronterface*.

Enfim, graças a este experimento verificou-se a possibilidade de utilizar o diâmetro de 1,75mm na impressora **3D** aprimorando a flexibilidade de operação deste equipamento.

Além disso, conforme apresentado, foram elaborados perfis de impressão para cada filamento



Figura 3.31: Teste com filamento *PVA* da marca Esun.

com coloração, translucidez e características distintas, sendo possível observar que certos filamentos de difícil impressão tornaram-se viáveis graças a ajustes de parâmetros do *slic3r*.

Capítulo 4

Conclusões

Este relatório apresentou diversos projetos efetuados na área de impressão **3D**.

A intenção foi a de aprimorar o conhecimento na área, resolver os principais óbices da produção na área de projetos de desenvolvimento e, principalmente, a de desenvolver melhorias neste setor, ainda tão futurístico e promissor.

Pelos resultados apresentados, os principais objetivos foram alcançados e muito conhecimento foi adquirido. Tanto da área de *software* quanto a de *hardware* da impressora **3D**.

As principais vitórias apresentadas neste relatório técnico se concentram na área de extrusão dupla e na de flexibilidade de utilização de filamentos de $1,75\text{mm}$ na impressora *Metamáquina 2*.

Estas inovações abriram novas oportunidades tecnológicas de implementação de melhorias no modelo atual, permitindo maior velocidade de impressão, leveza na parte móvel (carro da impressora), e flexibilidade de utilização.

As melhorias apresentadas são fruto de várias tentativas e erros realizados experimentalmente, sendo que as mais promissoras foram implementadas e outras, talvez, sejam utilizadas posteriormente.

A *Metamáquina* apresenta-se com estrutura e capacidade de produzir impressoras **3D** de fácil utilização pelo usuário, de boa qualidade de peças plásticas elaboradas e de grande flexibilidade de materiais e diâmetros de filamentos utilizados.

Ainda há muito a se desenvolver e construir neste ramo do conhecimento e muitas áreas de utilização desta tecnologia a serem descobertas e aprimoradas.

Este relatório apresentou algumas melhorias e incorporou mais conhecimento a esta área.

4.1 Considerações Finais

A área de desenvolvimento de novas tecnologias de impressão **3D** tem sido cada vez mais presente em diversos setores do conhecimento. Muito se tem a inovar e experimentar.

Durante um período de sete meses foram desenvolvidos diversos trabalhos na área de pesquisa e desenvolvimento, que abrangeu a tecnologia de extrusão dupla e melhorias no modelo atual da impressora produzida pela *Metamáquina*.

Todo o conhecimento aprendido foi passado a este relatório técnico, consistindo em inovações, desafios enfrentados e projetos desse setor.

Enfim, espera-se que estas melhorias continuem a ser efetuadas e que o empenho na elaboração deste relatório abra espaços para surgirem mais ideias e aprimoramentos nesta área.

4.2 Sugestões para Pesquisas Futuras

Para a área de pesquisas futuras consideram-se os itens apresentados a seguir para implementações na *Metamáquina 2*.

Como sugestões para pesquisas futuras pode-se citar um estudo de *design*, com novos materiais a serem utilizados para a carcaça da impressora, mais leves, precisos, belos esteticamente e de aproveitamento de espaço.

Também enfatiza-se a importância de se obter uma máquina de fácil montagem. De possuir acessos para possíveis manutenções. Utilizar micro conexões para permitir trocas rápidas de cabos de resistores e termistores, por exemplo.

Cita-se, também, a elaboração de uma ferramenta de impressão à distância, utilizando comunicação via redes sem fio (*wireless*), ou até mesmo via celular e acompanhamento em tempo real da impressão.

Sugere-se investir na inovação de maior precisão de movimentos e da utilização de extrusores mais finos, que permitem peças com maior riqueza de detalhes e, portanto, maior qualidade.

Para implementações das áreas que mais se apresentam como consumidoras da impressão **3D** está na área de design e arquitetura na elaboração de peças e maquetes de elevada complexidade para fazer testes, como por exemplo, em túnel de vento.

Outra área em que existem muitas linhas de pesquisas é a médica, focada na recuperação de tecidos de pacientes com danos teciduais severos e na impressão de órgãos para transplante utilizando células do próprio paciente.

Em geral, ainda há muitos ramos para elaborações de pesquisas neste setor, que está em fase de grande expansão tanto tecnológica quanto de áreas a serem descobertas, portanto, ainda há grande prosperidade para este ramo.

Apêndice A

Scripts dos Desenhos do OpensCAD

Neste apêndice serão apresentados os *scripts* dos desenhos dos modelos realizados em 3D.

Arquivos faltantes podem ser encontrados no sítio: <https://github.com/Metamaquina/Metamaquina2>

A.1 Porta Rolo de Filamento

```
1 // (c) 2013 Metamáquina <http://www.metamaquina.com.br/>
2 //
3 // Author:
4 // * Felipe C. da S. Sanches <fsanches@metamaquina.com.br>
5 // * Sara Rodriguez <sara@metamaquina.com.br>
6 //
7 // This program is free software: you can redistribute it and/or modify
8 // it under the terms of the GNU General Public License as published by
9 // the Free Software Foundation, either version 3 of the License, or
10 // (at your option) any later version.
11 //
12 // This program is distributed in the hope that it will be useful,
13 // but WITHOUT ANY WARRANTY; without even the implied warranty of
14 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15 // GNU General Public License for more details.
16 //
17 // You should have received a copy of the GNU General Public License
18 // along with this program. If not, see <http://www.gnu.org/licenses/>.
19
20 // utils
21 use <rounded_square.scad>;
22 use <tslot.scad>;
23 include <render.h>;
24 include <BillOfMaterials.h>;
25 use <mm2logo.scad>;
26 use <domed_cap_nuts.scad>;
27
28 //measures
29 thickness = 6;
30 margin = 10;
31
32 feet_height = 12;
33 feet_width = 50;
34 base_height = 80;
35
36 adjust=12;
37
38 total_width = 160+2*thickness+2*margin+adjust/2;
39
40 top_cut_height = 20;
41 top_cut_width = 9;
42
```

```

43 radius = 15;
44 rad=3/2;
45
46 spool_holder_width = 160+2*margin+2*thickness;
47 spool_holder_height = 40;
48
49 tslot_length = 16;
50 tslot_diameter = 3;
51
52 hole_domed_cap_nut=12.75;
53
54 bar_diameter=8;
55 bar_length= spool_holder_width+hole_domed_cap_nut*2;
56
57 total_height = 160/2+35/2-bar_diameter/2+top_cut_height+feet_height+
      top_cut_width/2;
58 radius_feet=5;
59
60 diameter=3;
61
62 //TSLOTS
63 sidepanel_TSLOTS = [
64 //parameters => [x, y, width, angle]
65 [(total_width/2-thickness),15, 50, 0],
66 [-(total_width/2-thickness),15, 50, 0],
67 ];
68
69 front_and_back_panels_TSLOT_SHAPES = [
70 //parameters => [x, y, angle]
71 [tslot_diameter,tslot_length,0],
72 ];
73
74 //side panel
75 module FilamentSpoolHolder_sidepanel_face(){
76
77   difference(){
78     union(){
79       hull()
80       for (i=[-1,1]){
81         translate([i*40,total_height-radius])
82         circle(r=radius);
83
84         translate([i*(total_width/2-radius),base_height])
85         circle(r=radius);
86
87         translate([i*total_width/2,feet_height])
88         circle(r=0.1);
89       }
90
91       for (i=[-1,1]){
92         translate([i*(total_width-feet_width)/2,(feet_height+radius_feet)/2])
93         rounded_square([feet_width,feet_height+radius_feet], corners=[radius_feet,
94           radius_feet, radius_feet, radius_feet], center=true);
95       }
96
97     union(){
98       translate([0,(total_height-top_cut_height/2)])
99       square([top_cut_width,top_cut_height],center = true);
100
101      translate([0,(total_height-top_cut_height)])
102      circle(r=top_cut_width/2);
103
104    //logo
105      translate([-3,(base_height*0.75)])

```

```

106      scale(5) mm_logo();
107
108 //tslots
109   for (i=[-1,1]){
110     translate([i*(total_width/2-thickness),15])
111     TSlot_holes();
112   }
113 }
114 }
115 }
116
117 //front and back panels
118 module FilamentSpoolHolder_front_and_back_panels_face(){
119   difference(){
120     union(){
121       translate([0,(base_height)/2])
122       square([spool_holder_width-2*thickness,spool_holder_height],center = true)
123       ;
124       //tslots joints
125       for (i=[-1,1]){
126         translate([i*(spool_holder_width/2-thickness/2),15])
127         TSlot_joints(50);
128       }
129     }
130
131   for (i=[-1,1]){
132     translate([i*(spool_holder_width/2+tslot_length),(base_height/2-i*
133       tslot_diameter)])
134     rotate([0,0,i*90])
135     tslot_shapes_from_list(front_and_back_panels_TSLOT_SHAPES);
136   }
137 }
138
139 //bar
140 module FilamentSpoolHolder_bar_face(){
141   circle(r=bar_diameter/2,center=true);
142 }
143
144 //spool
145 module FilamentSpoolHolder_spool_face(){
146   difference(){
147     circle(r=160/2,center=true);
148
149     circle(r=35/2,center=true);
150   }
151 }
152
153 module FilamentSpoolHolder_sidepanel_sheet(){
154   BillOfMaterials(category="Lasercut wood", partname="Filament Spool Holder Side
155   Panel");
156   material("lasercut")
157   linear_extrude(height=thickness)
158   FilamentSpoolHolder_sidepanel_face();
159   tslot_parts_from_list(sidepanel_TSLOTS);
160 }
161
162 module FilamentSpoolHolder_front_and_back_panels_sheet(){
163   BillOfMaterials(category="Lasercut wood", partname="Filament Spool Holder Side
164   Panel");
165   material("lasercut")
166   linear_extrude(height=thickness)
167   FilamentSpoolHolder_front_and_back_panels_face();
168 }
```

```

167
168 module FilamentSpoolHolder_bar_sheet(){
169   BillOfMaterials( str ("M8x" , "mm Threaded Rod") , 2 );
170   material("threaded metal")
171   linear_extrude( height=bar_length )
172   FilamentSpoolHolder_bar_face();
173 }
174
175 module FilamentSpoolHolder_sidepanels(){
176   rotate([0,0,90]){
177     translate([0, -spool_holder_width/2+thickness])
178     rotate([90,0,0])
179     FilamentSpoolHolder_sidepanel_sheet();
180
181     translate([0, spool_holder_width/2-thickness])
182     rotate([90,0,180])
183     FilamentSpoolHolder_sidepanel_sheet();
184   }
185 }
186
187 module FilamentSpoolHolder_front_and_back_panels(){
188   translate([0, -1*(total_width/2-thickness/2-adjust/2)])
189   rotate([90,0,0])
190   FilamentSpoolHolder_front_and_back_panels_sheet();
191
192   translate([0, 1*(total_width/2-thickness/2-adjust/2)])
193   rotate([90,0,180])
194   FilamentSpoolHolder_front_and_back_panels_sheet();
195 }
196
197 module FilamentSpoolHolder_bar_subassembly(){
198   translate([- (bar_length)/2,0,(total_height-top_cut_height-(top_cut_width-
199     bar_diameter)/2)])
200   rotate([0,90,0])
201   FilamentSpoolHolder_bar_sheet();
202
203   union(){
204     translate([1*((bar_length-2*hole_domed_cap_nut)/2),0,(total_height-
205       top_cut_height-(top_cut_width-bar_diameter)/2)])
206     rotate([0,90,0])
207     M8_domed_cap_nut();
208
209     translate([-1*((bar_length-2*hole_domed_cap_nut)/2),0,(total_height-
210       top_cut_height-(top_cut_width-bar_diameter)/2)])
211     rotate([0,270,0])
212     M8_domed_cap_nut();
213   }
214 }
215
216 module FilamentSpoolHolder(){
217   FilamentSpoolHolder_sidepanels();
218   FilamentSpoolHolder_front_and_back_panels();
219   FilamentSpoolHolder_bar_subassembly();
220 }
221
222 module FilamentSpool(){
223   BillOfMaterials(partname="Filament Spool");
224   translate([-160/2,0,total_height - top_cut_height - 35/2 + bar_diameter/2 - (
225     top_cut_width-bar_diameter)/2])
226   rotate([0,90,0])
227   material("ABS")
228   linear_extrude( height=160)
229   difference(){
230     circle(r=160/2);
231     circle(r=35/2);

```

```

228 }
229 }
230
231 FilamentSpoolHolder() ;
232 FilamentSpool() ;

```

A.2 Teste de Barras

```

1 // (c) 2013 Metamáquina <http://www.metamaquina.com.br/>
2 //
3 // Author:
4 // * Felipe C. da S. Sanches <fsanches@metamaquina.com.br>
5 //
6 // This program is free software: you can redistribute it and/or modify
7 // it under the terms of the GNU General Public License as published by
8 // the Free Software Foundation, either version 3 of the License, or
9 // (at your option) any later version.
10 //
11 // This program is distributed in the hope that it will be useful,
12 // but WITHOUT ANY WARRANTY; without even the implied warranty of
13 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 // GNU General Public License for more details.
15 //
16 // You should have received a copy of the GNU General Public License
17 // along with this program. If not, see <http://www.gnu.org/licenses/>.
18
19 include <Metamaquina2.h>;
20 include <BillOfMaterials.h>;
21
22 extruder_wiring_radius = 6;
23 YEndstopHolder_distance = 66;
24
25 // utils
26 //use <utils.scad>;
27 //use <mm2logo.scad>;
28 use <rounded_square.scad>;
29 use <tslot.scad>;
30
31 //sub assemblies
32 include <endstop.h>;
33 include <heated_bed.h>;
34 use <laserCut_extruder.scad>;
35 use <RAMBo.scad>;
36 use <jhead.scad>;
37
38 //parts
39 include <NEMA.h>;
40 include <coupling.h>;
41 include <washers.h>;
42 include <bolts.h>;
43 include <nuts.h>;
44 include <spacer.h>;
45 include <lm8uu_bearing.h>;
46 include <jhead.h>;
47 include <PowerSupply.h>;
48 use <608zz_bearing.scad>;
49 use <domed_cap_nuts.scad>;
50 use <belt-clamp.scad>;
51 use <cable_clips.scad>;
52
53 //3d printed parts

```

```

54 include <ZLink.h>;
55 use <bar-clamp.scad>;
56
57 //include <Metamaquina2.scad>
58
59 rods_radius_clearance = 0.04; //extra room for the X and Z rods
60
61 //For the actual build volume we avoid using the marginal
62 //region around the heated bed
63
64 HeatedBed_X = BuildVolume_X + 15; // 215 mm
65 HeatedBed_Y = BuildVolume_Y + 20; // 220 mm
66
67 hack_coupleings = 5; // for astethical purposes, the z-couplings are animated
68     //rotating <hack_coupleings> times slower than the correct mechanical behaviour
69 time = $t;
70 function carx_demo(time) = sin(360*time*7)*BuildVolume_X/2;
71 function cary_demo(time) = cos(360*time*7)*BuildVolume_Y/2;
72 //function carz_demo(time) = (0.5+0.5*sin(360*time))*0.3*BuildVolume_Y/2 + 0.7*
73     //BuildVolume_Y/2;
74 function carz_demo(time) = time*BuildVolume_Z;
75 function coupling_demo(time) = (360*carz_demo(time)/1.25)/hack_coupleings;
76
77 /* Positioning of the extruder assembly */
78 XCarPosition = -100; //carx_demo(time);
79 YCarPosition = 0; //cary_demo(time);
80 ZCarPosition = 150; //carz_demo(time);
81 //-----
82
83 //machine configs:
84
85 /* whether or not to add holes for a PowerSupply manufactured by Hiqua and sold
86 by Nodaji in Brazil */
87 HQUA_POWERSUPPLY=true;
88
89 /* dimensions of the machine feet */
90 feetwidth = 50;
91 feetheight = 12;
92
93 /*Here are a bunch of constants that determine the overall positioning
94 and dimensions of the several acrylic/plywood panels:*/
95
96 BuildPlatform_SidePanels_distance = 40;
97 SidePanels_distance = HeatedBed_X + 2*BuildPlatform_SidePanels_distance;
98
99 RightPanel_baseheight = 92;
100 RightPanel_basewidth = 2*(HeatedBed_Y)+10;
101
102 //Z_rods_distance = 388; //PrusaAir2
103 Z_rods_distance = SidePanels_distance + 2*(z_rod_z_bar_distance + NEMA17_width/2
104     + 5);
105 //TODO: machine_width = ?;
106 machine_height = BuildVolume_Z + 207.2; //why?
107
108 XZStage_offset = 20;
109 XZStage_position = RightPanel_basewidth/2 + XZStage_offset;
110 z_max_endstop_x = XZStage_position - 41;
111 z_max_endstop_y = machine_height - 19;
112
113 z_min_endstop_x = z_max_endstop_x - 28;
114 z_min_endstop_y = 109;
115

```

```

116 baseh = 35;
117 ArcPanel_rear_advance = 105;
118 horiz_bars_length = SidePanels_distance + 2*(m8_nut_height + m8_washer_thickness
    );
119 base_bars_height = 17;
120 base_bars_Zdistance = 50;
121
122 bar_cut_length=13;
123 Y_rod_length = RightPanel_basewidth - 2*(bar_cut_length + m8_diameter/2) + 24;
124 Y_rod_height = base_bars_Zdistance + base_bars_height + 10.2;//TODO
125 BottomPanel_width=60;
126 Z_rod_sidepanel_distance = (Z_rads_distance - SidePanels_distance)/2 + thickness
    ;
127
128 heatedbed_spring_length = 13; //TODO:
129 heatedbed_spring_compressed_length = 7.4; //TODO:
130 compressed_spring=1;
131
132 YPlatform_height = Y_rod_height + lm8uu_diameter/2;
133 pcb_height = YPlatform_height + thickness +
134 heatedbed_spring_compressed_length*compressed_spring + heatedbed_spring_length
    *(1-compressed_spring);
135
136 BuildPlatform_height = pcb_height + heated_bed_pcb_thickness +
    heated_bed_glass_thickness;
137
138 //machine_x_dim is the actual width of the whole machine
139 machine_x_dim = Z_rads_distance+2*(lm8uu_diameter/2+thickness);
140
141 XEnd_extra_width = 30;
142 XEnd_box_size = lm8uu_diameter/2 + z_rod_z_bar_distance + ZLink_rod_height;
143
144 //height of the bottom panel acrylic/plywood sheet:
145 BottomPanel_zoffset = feethight + NEMA17_length + 2;
146
147 Z_rod_length = machine_height - BottomPanel_zoffset + thickness;
148 Z_bar_length = thickness + machine_height - BottomPanel_zoffset -
    motor_shaft_length;
149
150 margin=4;
151 tslot_extra=thickness+margin; //todo
152 XPlatform_width = X_rads_distance + X_rads_diameter + 2*margin + 2* tslot_extra;
153 XEnd_width = XPlatform_width+XEnd_extra_width;
154 num_extruders = 1;
155 extra_extruder_length = 50; //TODO
156 XCarriage_padding = 6;
157 XCarriage_nozzle_hole_radius = 20;
158 XCarriage_width = XPlatform_width + 22;
159 //XCarriage_width = XPlatform_width;
160 //XCarriage_width = XEnd_width;
161 XCarriage_length = 82 + (num_extruders-1) * extra_extruder_length;
162 XCarriage_lm8uu_distance = XCarriage_length - 30;
163
164 nozzle_hole_width = 50;
165 nozzle_hole_length = machine_x_dim - 2*XEnd_box_size - nozzle_hole_width - 2*
    thickness - 2*20;
166
167 belt_offset = 26;
168 belt_width=5;
169 belt_clamp_height = 9;
170 PulleyRadius = 6;
171 IdlerRadius = 11;
172 XMotor_height = 31;
173 XIdler_height = XMotor_height + PulleyRadius - IdlerRadius;
174 X_rod_length = machine_x_dim - 2*thickness;

```

```

175 X_rod_height = XMotor_height + PulleyRadius - lm8uu_diameter/2 - 2*thickness;
176
177 XCarriage_height = thickness + X_rod_height + lm8uu_diameter/2;
178
179 nozzle_tip_distance = jhead_length-jhead_instalation_depth - thickness -
    XCarriage_height;
180 echo(str("nozzle_tip_distance:", nozzle_tip_distance));
181
182 RightPanel_backwidth = 55;
183 RightPanel_backheight = machine_height - RightPanel_baseheight;
184
185 rear_backtop_advance = XZStage_position - (XPlatform_width/2 + XEnd_extra_width
    + 10) - RightPanel_backwidth;
186
187 RightPanel_topheight = 30;
188 RightPanel_topwidth = XZStage_position + 30 - rear_backtop_advance;
189
190 ArcPanel_width = SidePanels_distance - 2 * thickness;
191 ArcPanel_height = 140; //TODO: make it depend on the machine height
192
193 //code modified below
194
195 SidePanel_TSLOT_SHAPES = [
196 //parameters => [x, y, angle]
197 [rear_backtop_advance+RightPanel_topwidth-25-5, machine_height+thickness,
    180],
198 [rear_backtop_advance+RightPanel_topwidth-25-5, machine_height+thickness
    -309.5, 0],
199 ];
200
201 TopPanel_TSLOTS = [
202 //parameters => [x, y, width, angle]
203 [Z_rods_distance/2-Z_rod_sidepanel_distance + thickness/2, 0, 0, 0],
204 ];
205
206 BottomPanel_TSLOTS = [
207 //parameters => [x, y, width, angle]
208 [Z_rods_distance/2-Z_rod_sidepanel_distance + thickness/2, 0, 0, 0],
209 ];
210
211 module calibracao(){
212   import("calibracao.dxf");
213 }
214
215 // 2d shapes for laser-cutting:
216
217 module RodEndTop_face(){
218 union(){
219   translate([30,0,0]){
220     RodEnd_face(z_rod_z_bar_distance+8);
221     translate([-30,0,0])
222     RodEnd_face(0, third_hole=false);
223   }
224 }
225 }
226
227 module RodEndBottom_face(){
228 union(){
229   translate([0,0,0]){
230     RodEnd_face(0, third_hole=false);
231     translate([-30,0,0])
232     RodEnd_face(0, third_hole=false);
233   }
234 }
235 }

```

```

236
237 //!MachineRightPanel_face();
238 module MachineRightPanel_face() {
239   difference() {
240     union() {
241       MachineSidePanel_plainface();
242
243       //tslots for top panel
244       translate([270,57,0])
245       rotate([0,0,90])
246       TSlot_joints();
247
248       //tslots for bottom panel
249       translate([270,56.9+303.3,0])
250       rotate([0,0,90])
251       TSlot_joints();
252     }
253
254     union() {
255       tslot_shapes_from_list(SidePanel_TSLOT_SHAPES);
256
257       translate([245,57+40,0])
258       circle(r = 2.9*2.54/2); //M5
259
260       translate([245,56.9+303.3-40,0])
261       circle(r = 2.9*2.54/2);
262     }
263   }
264 }
265
266 //!MachineSidePanel_plainface();
267 module MachineSidePanel_plainface() {
268   r1=0.1;
269   r2=60;
270   H=150;
271   k=19;
272
273   difference() {
274     //back
275     translate([rear_backtop_advance+k+90-1, RightPanel_baseheight-35])
276     square([60, RightPanel_backheight+35]);
277
278     translate([rear_backtop_advance+k+114, RightPanel_baseheight+72])
279     rotate([180,0,90])
280     scale(0.5) calibracao();
281   }
282 }
283
284 module TopPanel_holes() {
285   translate([Z_rods_distance/2,0]) {
286     //holes for Zrod and Zbar
287     circle(r = m8_diameter/2 + rods_radius_clearance);
288
289     translate([30,0])
290     circle(r = m8_diameter/2 + rods_radius_clearance);
291
292     translate([8, 0])
293     M3_hole();
294
295     translate([-z_rod_z_bar_distance - 8, -8])
296     M3_hole();
297
298     translate([-z_rod_z_bar_distance - 8, 8])
299     M3_hole();
300 }
```

```

301     translate([-z_rod_z_bar_distance - 8+68, -8])
302     M3_hole();
303
304     translate([-z_rod_z_bar_distance - 8+68, 8])
305     M3_hole();
306
307     translate([-59, -25,0])
308 //TSlot_holes(width=XEnd_box_size);
309 TSlot_holes(width=50);
310
311     translate([-z_rod_z_bar_distance,0]) {
312         //This hole's diameter is considerably larger than the threaded rod
313         //diameter
314         //in order to allow slightly bent rods to freely move. Otherwise, we
315         //would potentially have more whobble as a result of a tightly fixed rod.
316         circle(r=(m8_diameter+4)/2);
317     }
318
319 //!MachineTopPanel_face();
320 module MachineTopPanel_face() {
321     sidewidth=78;
322     difference() {
323         translate([machine_x_dim/2 - sidewidth,-30])
324         rounded_square([sidewidth+45, 60], corners=[0, 30, 0, 30]);
325
326         TopPanel_holes();
327     }
328 }
329
330 module BottomPanel_holes(){
331     //holes for Z rods
332     translate([Z_rods_distance/2,0]){
333         circle(r=m8_diameter/2 + rods_radius_clearance);
334
335         translate([30,0])
336         circle(r = m8_diameter/2 + rods_radius_clearance);
337
338         translate([30, -8]) M3_hole();
339         translate([30, 8]) M3_hole();
340
341         translate([0, -8]) M3_hole();
342         translate([0, 8]) M3_hole();
343     }
344
345     //holes for ZMotors
346     translate([Z_rods_distance/2 - z_rod_z_bar_distance, 0])
347     NEMA17_holes(r=27/2); //This should be large enough to let the coupling pass
348     through the hole
349
350     rotate([0,0,0])
351     translate([Z_rods_distance/2 - Z_rod_sidepanel_distance + thickness/2, -25,0])
352 //TSlot_holes(width=XEnd_box_size);
353     TSlot_holes(width=50);
354
355 //!MachineBottomPanel_face();
356 module MachineBottomPanel_face() {
357     render(){
358         difference(){
359             translate([(machine_x_dim/2-78),-30])
360             rounded_square([78+45, 60], corners=[0, 30, 0, 30]);
361
362             BottomPanel_holes();

```

```

363     }
364   }
365 }
366
367 // 3d preview of lasercut plates:
368
369 module RodEnd_ZTopRight_sheet() {
370   BillOfMaterials(category="Lasercut wood", partname="RodEnd Z Top Right");
371
372   translate([Z_rods_distance/2+30, -XZStage_offset, machine_height+thickness])
373   rotate([0,0,180])
374   RodEndTop_sheet();
375 }
376
377 module RodEnd_ZBottomRight_sheet() {
378   BillOfMaterials(category="Lasercut wood", partname="RodEnd Z Bottom Right");
379   translate([Z_rods_distance/2, -XZStage_offset, BottomPanel_zoffset - thickness
380   ])
381   rotate([0,0,180])
382   RodEndBottom_sheet();
383 }
384
385 module RodEndTop_sheet() {
386   {//TODO: Add these parts to the CAD model
387     BillOfMaterials("M3x25 bolt", 3);
388     BillOfMaterials("M3 washer", 3);
389     BillOfMaterials("M3 lock-nut", 3);
390   }
391
392   material("lasercut")
393   linear_extrude(height=thickness)
394   RodEndTop_face();
395 }
396
397 module RodEndBottom_sheet() {
398   {//TODO: Add these parts to the CAD model
399     BillOfMaterials("M3x20 bolt", 2);
400     BillOfMaterials("M3 washer", 2);
401     BillOfMaterials("M3 lock-nut", 2);
402   }
403
404   material("lasercut")
405   linear_extrude(height=thickness)
406   RodEndBottom_face();
407 }
408
409 module MachineRightPanel_sheet() {
410   BillOfMaterials(category="Lasercut wood", partname="Machine Right Panel");
411
412   translate([SidePanels_distance/2, RightPanel_basewidth/2, 0])
413   rotate([0,0,-90])
414   rotate([90,0,0]){
415     material("lasercut")
416     linear_extrude(height=thickness)
417     MachineRightPanel_face();
418   }
419 }
420
421 module MachineTopPanel_sheet() {
422   BillOfMaterials(category="Lasercut wood", partname="Machine Top Panel");
423
424   {//TODO: Add these parts to the CAD model
425     BillOfMaterials("M3x25 bolt", 2);
426     BillOfMaterials("M3 washer", 2);

```

```

427     BillOfMaterials( "M3 lock-nut" , 2 ) ;
428 }
429
430 translate([0,-XZStage_offset,machine_height]) {
431     material("lasercut")
432     linear_extrude(height=thickness)
433     MachineTopPanel_face();
434
435     tslot_parts_from_list(TopPanel_TSLOTS);
436 }
437 }
438
439 module MachineBottomPanel_sheet() {
440     BillOfMaterials(category="Lasercut wood", partname="Machine Bottom Panel");
441
442     translate([0,-XZStage_offset,BottomPanel_zoffset]) {
443         material("lasercut")
444         linear_extrude(height=thickness)
445         MachineBottomPanel_face();
446
447         translate([0,0,6])
448         rotate([180,0,0])
449         tslot_parts_from_list(BottomPanel_TSLOTS);
450     }
451 }
452
453 module Z_couplings() {
454     translate([machine_x_dim/2 - thickness - lm8uu_diameter/2 -
455             z_rod_z_bar_distance, -XZStage_offset, BottomPanel_zoffset +
456             motor_shaft_length - coupling_shaft_depth])
457     coupling();
458 }
459
460 module ZRods() {
461     BillOfMaterials(str("M8x",Z_rod_length,"mm Smooth Rod"), 2);
462
463     material("metal") {
464         translate([machine_x_dim/2 - thickness - lm8uu_diameter/2, -XZStage_offset,
465                 BottomPanel_zoffset])
466         cylinder(r=8/2, h=Z_rod_length);
467
468     }
469
470 module ZBars() {
471     BillOfMaterials(str("M8x",Z_bar_length,"mm Threaded Rod"), 2);
472
473     material("threaded metal") {
474         translate([machine_x_dim/2 - thickness - lm8uu_diameter/2 -
475                 z_rod_z_bar_distance, -XZStage_offset, BottomPanel_zoffset +
476                 motor_shaft_length])
477         cylinder(r=m8_diameter/2, h=Z_bar_length);
478     }
479
480 module LaserCutPanels() {
481     MachineTopPanel_sheet();
482     MachineRightPanel_sheet();
483     MachineBottomPanel_sheet();
484     RodEnd_ZTopRight_sheet();
485     RodEnd_ZBottomRight_sheet();
486 }
```

```

486
487 module ZMotors() {
488   translate([Z_rods_distance/2 - z_rod_z_bar_distance, -XZStage_offset ,
489             BottomPanel_zoffset])
490   rotate([180,0,0]) rotate(90) NEMA17_subassembly();
491 }
492 module ZAxis() {
493   ZMotors();
494   ZRods();
495   ZBars();
496   Z_couplings();
497 }
498
499 //!LaserCutPanels();
500 module CalibrationZBars() {
501   LaserCutPanels();
502   ZAxis();
503 }
504
505 CalibrationZBars();

```

A.3 Régulas de Calibração

```

1 //
2 // Author:
3 // * Sara Rodriguez <sara@metamaquina.com.br>
4 //
5 // This program is free software: you can redistribute it and/or modify
6 // it under the terms of the GNU General Public License as published by
7 // the Free Software Foundation, either version 3 of the License, or
8 // (at your option) any later version.
9 //
10 // This program is distributed in the hope that it will be useful,
11 // but WITHOUT ANY WARRANTY; without even the implied warranty of
12 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 // GNU General Public License for more details.
14 //
15 // You should have received a copy of the GNU General Public License
16 // along with this program. If not, see <http://www.gnu.org/licenses/>.
17
18 include <Metamaquina2.h>;
19 include <BillOfMaterials.h>;
20
21 extruder_wiring_radius = 6;
22 YEndstopHolder_distance = 66;
23
24 // utils
25 use <utils.scad>;
26 use <mm2logo.scad>;
27 use <rounded_square.scad>;
28 use <tslot.scad>;
29 include <Metamaquina2.scad>
30
31 // dimensions
32 tolerancia=0;
33
34 module calibracao(){
35   import("calibracao.dxf");
36 }
37

```

```

38 module FrontRule_face(){
39     difference(){
40         union(){
41             translate([0,0])
42             square([283, 50], center=true);
43             translate([0,25+25/2])
44             rounded_square([283+30, 25], corners=[10,10,10,10], center=true);
45         }
46         union(){
47             translate([-44,15+16])
48             scale(0.5) calibracao();
49         }
50     for (i=[-1,1]){
51
52 // barras transversais
53     translate([i*Y_rods_distance/2,-11+tolerancia,0])
54     rounded_square([m8_diameter,50], corners=[0,0,m8_diameter/2,m8_diameter
55 /2], center=true);
56 // arruelas encostadas nas laterais
57     translate([i*(SidePanels_distance-2*thickness-m8_washer_thickness)/2,
58 m8_washer_D/2-50/2+tolerancia,0])
59     square([m8_washer_thickness,50], center=true);
60 // porcas encostadas nas laterais
61     translate([i*(SidePanels_distance-2*thickness-2*m8_washer_thickness-
62 m8_nut_height)/2,m8_nut_R-50/2+1.1+tolerancia,0])
63     square([m8_nut_height,50], center=true);
64 // arruelas encostadas nas laterais do rolamento
65     translate([i*(bearing_thickness+2*mudguard_washer_thickness+2*
66 washer_thickness+1.25)/2,m8_washer_D/2-50/2+tolerancia,0])
67     square([1.25,50], center=true);
68 // porcas encostadas nas laterais do rolamento
69     translate([i*(bearing_thickness+2*mudguard_washer_thickness+2*
70 washer_thickness+2*1.25+m8_nut_height)/2,m8_nut_R-50/2+1.1+tolerancia,0])
71     square([m8_nut_height,50], center=true);
72 }
73 //rolamento
74     translate([0,-8+tolerancia,0])
75     square([bearing_thickness+2*mudguard_washer_thickness+2*washer_thickness
76 ,50], center=true);
77 }
78 }
79
80 module FrontRule_sheet(){
81     material("lasercut")
82     linear_extrude(height=thickness)
83         FrontRule_face();
84 }
85
86 module FrontRule(){
87     translate([0,5,0])
88     translate([0, -RightPanel_basewidth/2 + bar_cut_length+thickness/2,
89 base_bars_Zdistance + base_bars_height])
89     rotate([90,0,0])
90     FrontRule_sheet();
91 }
92
93 module RearRule_face(){
94     difference(){
95         union(){

```

```

96     translate([0,0])
97     square([283, 50], center=true);
98     translate([0,25+25/2])
99     square([283, 25], center=true);
100    }
101    union(){
102    rotate([180,0,0])
103    rotate([180,180,0])
104    translate([-44,15+16])
105    scale(0.5) calibracao();
106
107    for (i=[-1,1]){
108
109 // barras transversais
110    translate([i*Y_rods_distance/2,-11+tolerancia,0])
111    rounded_square([m8_diameter,50], corners=[0,0,m8_diameter/2,
112 m8_diameter/2], center=true);
113 // arruelas encostadas nas laterais
114    translate([i*(SidePanels_distance-2*thickness-m8_washer_thickness)/2,
115 m8_washer_D/2-50/2+tolerancia,0])
116    square([m8_washer_thickness,50], center=true);
117 // porcas encostadas nas laterais
118    translate([i*(SidePanels_distance-2*thickness-2*m8_washer_thickness-
119 m8_nut_height)/2,m8_nut_R-50/2+1.1+tolerancia,0])
120    square([m8_nut_height,50], center=true);
121
122 // porca da direita encostada no rolamento
123    translate([(bearing_thickness+2*mudguard_washer_thickness+2*
124 washer_thickness+m8_nut_height)/2,m8_nut_R-50/2+1.1+tolerancia,0])
125    square([m8_nut_height,50], center=true);
126 // porca da esquerda encostada no rolamento
127    translate([-(-(bearing_thickness+2*mudguard_washer_thickness+2*
128 washer_thickness+thickness+2*thickness+2*m8_washer_thickness)/2,
129 m8_nut_R-50/2+1.1+tolerancia,0])
130    square([m8_nut_height,50], center=true);
131
132 // arruela encostada no rolamento
133    translate([-(-(bearing_thickness+2*mudguard_washer_thickness+2*
134 washer_thickness+2*thickness+m8_washer_thickness)/2,m8_washer_D
135 /2-50/2+tolerancia,0])
136    square([m8_washer_thickness,50], center=true);
137
138 // madeira motor
139    translate([-(-(bearing_thickness+2*mudguard_washer_thickness+2*
140 washer_thickness+thickness)/2,-14.5+tolerancia,0])
141    square([thickness,50], center=true);
142  }
143 }
144
145 module RearRule_sheet(){
146   material("lasercut")
147   linear_extrude(height=thickness)
148   RearRule_face();
149 }
150

```

```

151 module RearRule() {
152   translate([0 ,430 ,0])
153   translate([0 , -RightPanel_basewidth/2 + bar_cut_length+thickness/2 ,
154             base_bars_Zdistance + base_bars_height])
155   rotate([90 ,0 ,0])
156   RearRule_sheet();
157 }
158 FrontRule();
159 RearRule();

```

A.4 Prendedor de Extrusores

```

1 // Extruders clamp
2 // Used for joining 2 peeks
3 //
4 // Author:
5 // * Sara Rodriguez <sara@metamaquina.com.br>
6 //
7 // This program is free software: you can redistribute it and/or modify
8 // it under the terms of the GNU General Public License as published by
9 // the Free Software Foundation, either version 3 of the License, or
10 // (at your option) any later version.
11 //
12 // This program is distributed in the hope that it will be useful,
13 // but WITHOUT ANY WARRANTY; without even the implied warranty of
14 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15 // GNU General Public License for more details.
16 //
17 // You should have received a copy of the GNU General Public License
18 // along with this program. If not, see <http://www.gnu.org/licenses/>.
19
20 epsilon = 0.1;
21
22 include <bolts.h>
23 include <configuration.scad>
24 include <BillOfMaterials.h>;
25 include <render.h>
26 include <lm8uu_bearing.h>
27 include <nuts.h>;
28 include <spacer.h>;
29 use <Metamaquina2.scad>
30 use <utils.scad>;
31
32 module extrudersclamp_face(r = 11/2, folga = 0.5, h = 6, largura_total = 84,
33                     comprimento_total = 23, cover = false, abertura = 3, separador = 1, distancia
34                     = 26){
35   difference(){
36     union(){
37       for (i=[-1,1]){
38         //círculo maior para colocação dos parafusos
39         translate([i*distancia/2, 0])
40         circle(r = comprimento_total/2, $fn=80);
41
42       hull(){
43         //círculo menor - extremidades
44         translate([i*33,0])
45       }
46     }
47   }
48 }

```

```

47     circle(r = m4_diameter*3/2, $fn=80);
48
49     //círculo maior para colocação dos parafusos
50     translate([i*distancia/2, 0])
51     circle(r = comprimento_total/2, $fn=80);
52
53     //junção meio
54     translate([-distancia/2, -30/4])
55     square([distancia, 30/2]);
56   }
57 }
58 }
59
60
61 union(){
62   for (i=[-1,1]){
63     //furo que prende o extrusor
64     translate([i*distancia/2,0])
65     circle(r = r+folga/2, $fn=80);
66
67     //define profundidade do sulco para encaixe do extrusor
68     translate([i*distancia/2,0])
69     circle(r = r+folga/2, $fn=80);
70
71     //furo para encaixe dos extrusores
72     translate([i*distancia/2,0])
73     circle(r = r+folga/2, $fn=80);
74
75     //furo para parafusos que prendem no carrinho
76     translate([i*35,0])
77     circle(r = m3_diameter/2, $fn=80);
78
79   if(cover){
80     difference(){
81       circle(r = 50);
82       circle(r = 20-folga);
83     }
84   }
85 }
86 }
87 }
88 }
89
90 bearing_sandwich_spacing = 12;
91 thickness = 6;
92 XCarPosition = -100;
93 XMotor_height = 31;
94 PulleyRadius = 6;
95 XCarriage_padding = 6;
96 margin=4;
97 tslot_extra=thickness+margin;
98 X_rod_height = XMotor_height + PulleyRadius - lm8uu_diameter/2 - 2*thickness;
99 XCarriage_height = thickness + X_rod_height + lm8uu_diameter/2;
100 XCarriage_length = 82;
101 XCarriage_lm8uu_distance = XCarriage_length - 30;
102 XPlatform_width = X_rods_distance + X_rods_diameter + 2*margin + 2* tslot_extra;
103
104 module extrudersclamp(){
105   BillOfMaterials(category="Lasercut wood", partname="Extruders Clamp", ref(""));
106   material("lasercut")
107   linear_extrude(height=6)
108   extrudersclamp_face(cover=false);
109 }
110
111 module extrudersclamp_cover(){

```

```
112 BillOfMaterials (category="Lasercut wood", partname="Extruders Clamp", ref="" );
113 material("lasercut")
114 linear_extrude(height=6)
115 extrudersclamp_face(cover=true);
116 }
117
118 //descomente para ver no carrinho
119
120 color("red")
121 translate([0, -0.5, h/2])
122 translate([XCarPosition-0.4, 0, XCarriage_height-2*thickness]) {
123     translate([0,0,6])
124     extrudersclamp();
125     translate([0,0,12])
126     extrudersclamp_cover();
127 }
128 //XCarriage();
129
130 translate([XCarPosition, 0, XCarriage_height]) {
131     %XCarriage_bottom_sheet();
132 }
133
134
135 //descomente para ver apenas a peça
136 /*
137 translate([0, -0.5, h/2]) {
138 extrudersclamp();
139
140 translate([0,0,6])
141 extrudersclamp_cover();
142 }
143 */
```

Apêndice B

Scripts do Software Arduino IDE

Neste apêndice serão apresentados os *scripts* do teste de barras e de parte do *firmware* da impressora **3D** com dupla extrusão.

Arquivos faltantes podem ser encontrados no sítio: <https://github.com/Metamaquina/Repetier-Firmware>

B.1 Teste de Barras

Nesta seção serão apresentados os *scripts* para a rotação constante e unidirecional do motor e para a rotação gradual e bidirecional.

B.1.1 Rotação Unidirecional e Constante

```
1 /*
2 by Sara
3
4 round and round!
5 */
6
7 #include <Stepper.h>
8
9 const int steps_per_revolution = 10000;
10
11 int menable = 14; //megatronics 24
12 int mdir = 21; //megatronics 28
13 int mstep = 15; //megatronics 26
14
15 int button = 18; //megatronics 41
16 int button_read = 0;
17
18 Stepper motor(steps_per_revolution, mstep, mdir);
19
20 void setup()
21 {
22   motor.setSpeed(200);
23   Serial.begin(9600);
24
25   pinMode(button, INPUT);
26
27   pinMode(mstep, OUTPUT);
28   pinMode(mdir, OUTPUT);
29   pinMode(menable, OUTPUT);
30 }
31
32 void loop()
33 {
34   button_read = digitalRead(button);
35
36   if (button_read == 1) {
```

```

37     Serial.println("liga");
38     for (int i=0 ; i<=10; i++){
39         motor.step(steps_per_revolution);
40     }
41 }
42 }
```

B.1.2 Rotação Bidirecional e Gradual

```

1 /*
2 by Sara
3
4 round and round!
5 */
6
7 //This is an example of how you would control 1 stepper
8
9 #include <AccelStepper.h>
10
11 int motorSpeed = 9600; //maximum steps per second (about 3rps / at 16 microsteps
12
13 int motorAccel = 80000; //steps/second/second to accelerate
14 int motorDirPin = 28; //digital pin 2
15 int motorStepPin = 26; //digital pin 3
16
17 int motorEnable = 24;
18
19 int button = 41;
20 int button_read = 0;
21
22 //set up the accelStepper instance
23 //the "1" tells it we are using a driver
24 AccelStepper stepper(1, motorStepPin, motorDirPin);
25
26
27
28 void setup(){
29     pinMode(button, INPUT);
30
31     pinMode(motorDirPin, OUTPUT);
32     pinMode(motorStepPin, OUTPUT);
33     pinMode(motorEnable, OUTPUT);
34
35     stepper.setMaxSpeed(motorSpeed);
36     stepper.setSpeed(motorSpeed);
37     stepper.setAcceleration(motorAccel);
38
39     stepper.moveTo(32000); //move 32000 steps (should be 10 rev)
40 }
41
42 void loop(){
43
44     button_read = digitalRead(button);
45
46     if (button_read == 1) {
47         for (int i=0; i<=10000; i++){
48             //if stepper is at desired location
49             if (stepper.distanceToGo() == 0){
50                 //go the other way the same amount of steps
51                 //so if current position is 400 steps out, go position -400
```

```

52         stepper.moveTo(-stepper.currentPosition());
53     }
54     //these must be called as often as possible to ensure smooth operation
55     //any delay will cause jerky motion
56     stepper.run();
57 }
58 }
59 }
```

B.2 Arquivo configuration.h do Repetier-Firmware Adaptado para Dupla Extrusão

```

1 /*
2  This file is part of Repetier-Firmware.
3
4  Repetier-Firmware is free software: you can redistribute it and/or modify
5  it under the terms of the GNU General Public License as published by
6  the Free Software Foundation, either version 3 of the License, or
7  (at your option) any later version.
8
9  Repetier-Firmware is distributed in the hope that it will be useful,
10 but WITHOUT ANY WARRANTY; without even the implied warranty of
11 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 GNU General Public License for more details.
13
14 You should have received a copy of the GNU General Public License
15 along with Repetier-Firmware. If not, see <http://www.gnu.org/licenses/>.
16
17 */
18
19 #ifndef CONFIGURATION_H
20 #define CONFIGURATION_H
21
22 #define FIRMWARE_VERSION_STRING "RepetierFw MM2 2013-10-15\nhttps://github.com/Metamaquina/Repetier-Firmware/tree/MM2\_2013-10-15"
23 //#define FIRMWARE_VERSION_STRING "https://github.com/Metamaquina/Repetier-Firmware/tree/mm2"
24
25 /* Some words on units:
26
27 From 0.80 onwards the units used are unified for easier configuration, watch out
when transferring from older configs!
28
29 Speed is in mm/s
30 Acceleration in mm/s^2
31 Temperature is in degrees celsius
32
33
34 ##### IMPORTANT #####
35 ##          IMPORTANT          ##
36 ##### #####
37
38 For easy configuration, the default settings enable parameter storage in EEPROM.
39 This means, after the first upload many variables can only be changed using the
special
40 M commands as described in the documentation. Changing these value sin the
configuration.h
41 has no effect. Parameters overriden by EEPROM settings are calibartion values,
extruder
42 values except thermistor tables and some other parameter likely to change during
usage
```

```

43 like advance steps or ops mode.
44 To override EEPROM settings with config settings, set EEPROM_MODE 0
45
46 */
47
48
49 // BASIC SETTINGS: select your board type, thermistor type, axis scaling, and
   endstop configuration
50
51 //// The following define selects which electronics board you have. Please
   choose the one that matches your setup
52 // MEGA/RAMPS up to 1.2      = 3
53 // RAMPS 1.3/RAMPS 1.4      = 33
54 // Gen6                      = 5
55 // Gen6 deluxe                = 51
56 // Sanguinololu up to 1.1    = 6
57 // Sanguinololu 1.2 and above = 62
58 // Gen7 1.1 till 1.3.x       = 7
59 // Gen7 1.4.1 and later       = 71
60 // Teensylu (at90usb)        = 8 // requires Teensyduino
61 // Printrboard (at90usb)      = 9 // requires Teensyduino
62 // Folytyn 3D Master          = 12
63 // MegaTronics                = 70
64 // RUMBA                      = 80 // Get it from reprapdiscount
65 // Rambo                      = 301
66 // Arduino Due                 = 401 // This is only experimental
67
68 #define MOTHERBOARD 301
69 #include "pins.h"
70
71 // Uncomment the following line if oyu are using arduino compatible firmware
   made for Arduino version earlier then 1.0
72 // If it is incompatible you will get compiler errors about write functions not
   bicing compatible!
73 // #define COMPAT_PREI
74
75 /* Define the type of axis movements needed for your printer. The typical case
76 is a full cartesian system where x, y and z moves are handled by seperate motors
   .
77
78 0 = full cartesian system, xyz have seperate motors.
79 1 = z axis + xy H-gantry (x_motor = x+y, y_motor = x-y)
80 2 = z axis + xy H-gantry (x_motor = x+y, y_motor = y-x)
81 3 = Delta printers (Rostock, Kossel, RostockMax, Cerberus, etc)
82 Cases 1 and 2 cover all needed xy H gantry systems. If you get results mirrored
   etc. you can swap motor connections for x and y. If a motor turns in
83 the wrong direction change INVERT_X_DIR or INVERT_Y_DIR.
84 */
85 #define DRIVE_SYSTEM 0
86
87 // ##### Calibration #####
88 // ##                               ##
89 // ##### Calibration #####
90
91 /** Drive settings for the Delta printers
92 */
93 #if DRIVE_SYSTEM==3
94 /** | brief Delta drive type: 0 - belts and pulleys, 1 - filament drive
95 */
96 #define DELTA_DRIVE_TYPE 0
97
98 #if DELTA_DRIVE_TYPE == 0
99 /** | brief Pitch in mm of drive belt. GT2 = 2mm
100 */
101 #define BELT_PITCH 2

```

```

102
103 /** | brief Number of teeth on X, Y and Z tower pulleys
104 */
105 #define PULLEY_TEETH 20
106 #define PULLEY_CIRCUMFERENCE (BELT_PITCH * PULLEY_TEETH)
107
108 #elif DELTA_DRIVE_TYPE == 1
109
110 /** | brief Filament pulley diameter in milimeters
111 */
112 #define PULLEY_DIAMETER 10
113
114 #define PULLEY_CIRCUMFERENCE (PULLEY_DIAMETER * 3.1415927)
115
116 #endif
117
118
119 /** | brief Steps per rotation of stepper motor
120 */
121 #define STEPS_PER_ROTATION 400
122
123 /** | brief Micro stepping rate of X, Y and Z tower stepper drivers
124 */
125 #define MICRO_STEPS 16
126
127 /** | brief Number of delta moves in each line. Moves that exceed this figure
will be split into multiple lines.
128 Increasing this figure can use a lot of memory since 7 bytes * size of line
buffer * MAX_DELTA_SEGMENTS_PER_LINE
129 will be allocated for the delta buffer. With defaults 7 * 16 * 30 = 3360 bytes.
This leaves ~1K free RAM on an Arduino
130 Mega.
131 */
132 #define MAX_DELTA_SEGMENTS_PER_LINE 30
133
134 // Calculations
135 #define AXIS_STEPS_PER_MM ((float)(MICRO_STEPS * STEPS_PER_ROTATION) /
136 PULLEY_CIRCUMFERENCE)
136 #define XAXIS_STEPS_PER_MM AXIS_STEPS_PER_MM
137 #define YAXIS_STEPS_PER_MM AXIS_STEPS_PER_MM
138 #define ZAXIS_STEPS_PER_MM AXIS_STEPS_PER_MM
139
140 #else
141
142 /** Drive settings for printers with cartesian drive systems */
143
144 /** | brief Steps per rotation of stepper motor
145 */
146 #define STEPS_PER_ROTATION 200
147
148 /** | brief Micro stepping rate of X, Y and Z tower stepper drivers
149 */
150 #define MICRO_STEPS 16
151
152 /** | brief Pitch in mm of drive belt. GT2 = 2mm
153 */
154 #define BELT_PITCH 2
155
156 /** | brief Number of teeth on X, Y and Z tower pulleys
157 */
158 #define PULLEY_TEETH 16
159 #define PULLEY_CIRCUMFERENCE (BELT_PITCH * PULLEY_TEETH)
160
161 #define AXIS_STEPS_PER_MM ((float)(MICRO_STEPS * STEPS_PER_ROTATION) /
162 PULLEY_CIRCUMFERENCE)

```

```

162
163 /** | brief Number of steps for a 1mm move in x direction. Overridden if EEPROM
164 activated. */
165 #define XAXIS_STEPS_PER_MM AXIS_STEPS_PER_MM
166 /** | brief Number of steps for a 1mm move in y direction. Overridden if EEPROM
167 activated. */
168 #define YAXIS_STEPS_PER_MM AXIS_STEPS_PER_MM
169 /** | brief Number of steps for a 1mm move in z direction Overridden if EEPROM
170 activated. */
171 #define ZAXIS_STEPS_PER_MM (MICRO_STEPS * STEPS_PER_ROTATION / Z_THREAD_PITCH)
172
173 #endif
174
175 // ##### Extruder configuration #####
176 // ## Extruder configuration ##
177 // #####
178
179 /** Number of extruders. Maximum 2 extruder. */
180 #define NUM_EXTRUDER 2
181
182 #define EXT0_X_OFFSET 0
183 #define EXT0_Y_OFFSET 0
184 // for skeinforge 40 and later, steps to pull the plastic 1 mm inside the
extruder, not out. Overridden if EEPROM activated.
185 #define EX0_E_RATIO float(37)/float(11)
186 #define EX0_BOLT_DIAMETER 7.2
187 //#define FORMULA_EXT0_STEPS_PER_MM (MICRO_STEPS * STEPS_PER_ROTATION *
EX0_E_RATIO) / (3.1415 * EX0_BOLT_DIAMETER) // sara era descomentado
188 //#define EMPIRICAL_EXT0_STEPS_PER_MM 650 // sara era descomentado
189 #define EXT0_STEPS_PER_MM 130 // sara era EMPIRICAL_EXT0_STEPS_PER_MM 650 pode
ser 92 ou 92.5 calculado era 103
190 // What type of sensor is used?
191 // 1 is 100k thermistor (Epcos B57560G0107F000 - RepRap-Fab.org and many other)
: MM2 first thermistor batch
192 // 2 is 200k thermistor
193 // 3 is mendel-parts thermistor (EPCOS G550)
194 // 4 is 10k thermistor
195 // 5 is userdefined thermistor table 0 : SMD 15k thermistor for the heated bed
196 // 6 is userdefined thermistor table 1 : MM2 second Addtherm thermistor batch
197 // 7 is userdefined thermistor table 2 : MM2 third Addtherm thermistor batch
198 // 50 is userdefined thermistor table 0 for PTC thermistors
199 // 51 is userdefined thermistor table 0 for PTC thermistors
200 // 52 is userdefined thermistor table 0 for PTC thermistors
201 // 97 Generic thermistor table 1
202 // 98 Generic thermistor table 2
203 // 99 Generic thermistor table 3
204 // 100 is AD595
205 // 101 is MAX6675
206
207 // Overridden if EEPROM activated.
208 #define EXT0_TEMPSENSOR_TYPE 7 // sara era 1
209 // Analog input pin for reading temperatures or pin enabling SS for MAX6675
210 #define EXT0_TEMPSENSOR_PIN TEMP_0_PIN
211 // Which pin enables the heater
212 #define EXT0_HEATER_PIN HEATER_0_PIN
213 #define EXT0_STEP_PIN E0_STEP_PIN
214 #define EXT0_DIR_PIN E0_DIR_PIN
215 // set to false/true for normal / inverse direction
216 #define EXT0_INVERSE true
217 #define EXT0_ENABLE_PIN E0_ENABLE_PIN
218 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active
High) use 1

```

```

219 #define EXT0_ENABLE_ON false
220 // The following speed settings are for skeinforge 40+ where e is the
221 // length of filament pulled inside the heater. For repsnap or older
222 // skeinforge use higher values.
223 // Overridden if EEPROM activated.
224 #define EXT0_MAX_FEEDRATE 40 // sara era 12
225 // Feedrate from halted extruder in mm/s
226 // Overridden if EEPROM activated.
227 #define EXT0_MAX_START_FEEDRATE 10
228 // Acceleration in mm/s^2
229 // Overridden if EEPROM activated.
230 #define EXT0_MAX_ACCELERATION 3500 // sara era 1000
231 /** Type of heat manager for this extruder.
232 - 0 = Simply switch on/off if temperature is reached. Works always.
233 - 1 = PID Temperature control. Is better but needs good PID values. Defaults are
        a good start for most extruder.
234 Overridden if EEPROM activated.
235 */
236 #define EXT0_HEAT_MANAGER 1
237 /** Wait x seconds, after reaching target temperature. Only used for M109.
        Overridden if EEPROM activated. */
238 #define EXT0_WATCHPERIOD 1
239
240 /** | brief The maximum value, I-gain can contribute to the output.
241 A good value is slightly higher than the output needed for your temperature.
242 Values for starts:
243 130 => PLA for temperatures from 170–180°C
244 180 => ABS for temperatures around 240°C
245
246 The precise values may differ for different nozzle/resistor combination.
247 Overridden if EEPROM activated.
248 */
249 #define EXT0_PID_INTEGRAL_DRIVE_MAX 130
250 /** | brief lower value for integral part
251 The I state should converge to the exact heater output needed for the target
temperature.
252 To prevent a long deviation from the target zone, this value limits the lower
value.
253 A good start is 30 lower than the optimal value. You need to leave room for
cooling.
254 Overridden if EEPROM activated.
255 */
256 #define EXT0_PID_INTEGRAL_DRIVE_MIN 60
257 /** | brief P-gain. Overridden if EEPROM activated. */
258 #define EXT0_PID_P 24
259 /** | brief I-gain. Overridden if EEPROM activated.
260 */
261 #define EXT0_PID_I 0.88
262 /** | brief Dgain. Overridden if EEPROM activated.*/
263 #define EXT0_PID_D 80
264 // maximum time the heater is can be switched on. Max = 255. Overridden if
EEPROM activated.
265 #define EXT0_PID_MAX 255
266 /** | brief Faktor for the advance algorithm. 0 disables the algorithm.
Overridden if EEPROM activated.
267 K is the factor for the quadratic term, which is normally disabled in newer
versions. If you want to use
268 the quadratic factor make sure ENABLE_QUADRATIC_ADVANCE is defined.
269 L is the linear factor and seems to be working better than the quadratic
dependency.
270 */
271 #define EXT0_ADVANCE_K 0.0f
272 #define EXT0_ADVANCE_L 0.0f

```

```

275
276 /** | brief Temperature to retract filament when extruder is heating up.
277   Overridden if EEPROM activated.
277 */
278 #define EXT0_WAIT_RETRACT_TEMP    150
279 /** | brief Units (mm/inches) to retract filament when extruder is heating up.
280   Overridden if EEPROM activated. Set
280 to 0 to disable.
281 */
282 #define EXT0_WAIT_RETRACT_UNITS    0
283
284 /** You can run any gcode command son extruder deselect/select. Separate
285   multiple commands with a new line '\n'.
285 That way you can execute some mechanical components needed for extruder
285 selection or retract filament or whatever you need.
286 The codes are only executed for multiple extruder when changing the extruder. */
287 #define EXT0_SELECT_COMMANDS "M120 S5 P5\nM117 Extruder 1"
288 #define EXT0_DESELECT_COMMANDS ""
289 /** The extruder cooler is a fan to cool the extruder when it is heating. If you
289   turn the extruder on, the fan goes on. */
290 #define EXT0_EXTRUDER_COOLER_PIN -1
291 /** PWM speed for the cooler fan. 0=off 255=full speed */
292 #define EXT0_EXTRUDER_COOLER_SPEED 255
293
294
295 // ===== Configuration for second extruder
296
296 #define EXT1_X_OFFSET (26.5 * XAXIS_STEPS_PER_MM) // sara era 27
297 #define EXT1_Y_OFFSET (-1.4 * YAXIS_STEPS_PER_MM) // sara era -1.2
298 // for skeinforge 40 and later, steps to pull the plastic 1 mm inside the
298 // extruder, not out. Overridden if EEPROM activated.
299 #define EXT1_STEPS_PER_MM 130 // sara calculado era 103
300 // What type of sensor is used?
301 // 1 is 100k thermistor (Epcos B57560G0107F000 - RepRap-Fab.org and many other)
302 // 2 is 200k thermistor
303 // 3 is mendel-parts thermistor (EPCOS G550)
304 // 4 is 10k thermistor
305 // 5 is userdefined thermistor table 0
306 // 6 is userdefined thermistor table 1
307 // 7 is userdefined thermistor table 2
308 // 50 is userdefined thermistor table 0 for PTC thermistors
309 // 51 is userdefined thermistor table 0 for PTC thermistors
310 // 52 is userdefined thermistor table 0 for PTC thermistors
311 // 97 Generic thermistor table 1
312 // 98 Generic thermistor table 2
313 // 99 Generic thermistor table 3
314 // 100 is AD595
315 // 101 is MAX6675
316
317 // Overridden if EEPROM activated.
318 #define EXT1_TEMPSENSOR_TYPE 1
319 // Analog input pin for reading temperatures or pin enabling SS for MAX6675
320 #define EXT1_TEMPSENSOR_PIN TEMP_1_PIN
321 // Which pin enables the heater
322 #define EXT1_HEATER_PIN HEATER_1_PIN
323 #define EXT1_STEP_PIN E1_STEP_PIN
324 #define EXT1_DIR_PIN E1_DIR_PIN
325 // set to 0/1 for normal / inverse direction
326 #define EXT1_INVERSE true
327 #define EXT1_ENABLE_PIN E1_ENABLE_PIN
328 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active
328 // High) use 1
329 #define EXT1_ENABLE_ON false
330 // The following speed settings are for skeinforge 40+ where e is the
331 // length of filament pulled inside the heater. For repsnap or older

```

```

332 // skeinforge use either values .
333 // Overridden if EEPROM activated .
334 #define EXT1_MAX_FEEDRATE 40 // sara era 50
335 // Feedrate from halted extruder in mm/s
336 // Overridden if EEPROM activated .
337 #define EXT1_MAX_START_FEEDRATE 10 // sara era 12
338 // Acceleration in mm/s^2
339 // Overridden if EEPROM activated .
340 #define EXT1_MAX_ACCELERATION 3500 // sara era 10000
341 /** Type of heat manager for this extruder .
342 - 0 = Simply switch on/off if temperature is reached . Works always .
343 - 1 = PID Temperature control . Is better but needs good PID values . Defaults are
            a good start for most extruder .
344 Overridden if EEPROM activated .
345 */
346 #define EXT1_HEAT_MANAGER 1
347 /** Wait x seconds , after reaching target temperature . Only used for M109 .
            Overridden if EEPROM activated . */
348 #define EXT1_WATCHPERIOD 1
349
350 /** | brief The maximum value , I-gain can contribute to the output .
351
352 A good value is slightly higher then the output needed for your temperature .
353 Values for starts :
354 130 => PLA for temperatures from 170–180°C
355 180 => ABS for temperatures around 240°C
356
357 The precise values may differ for different nozzle/resistor combination .
358 Overridden if EEPROM activated .
359 */
360 #define EXT1_PID_INTEGRAL_DRIVE_MAX 130
361 /** | brief lower value for integral part
362
363 The I state should converge to the exact heater output needed for the target
            temperature .
364 To prevent a long deviation from the target zone , this value limits the lower
            value .
365 A good start is 30 lower then the optimal value . You need to leave room for
            cooling .
366 Overridden if EEPROM activated .
367 */
368 #define EXT1_PID_INTEGRAL_DRIVE_MIN 60
369 /** P-gain . Overridden if EEPROM activated . */
370 #define EXT1_PID_P 24
371 /** I-gain . Overridden if EEPROM activated .
372 */
373 #define EXT1_PID_I 0.88
374 /** Dgain . Overridden if EEPROM activated . */
375 #define EXT1_PID_D 80
376 // maximum time the heater is can be switched on . Max = 255 . Overridden if
            EEPROM activated .
377 #define EXT1_PID_MAX 255
378 /** | brief Faktor for the advance algorithm . 0 disables the algorithm .
            Overridden if EEPROM activated .
379 K is the factor for the quadratic term , which is normally disabled in newer
            versions . If you want to use
380 the quadratic factor make sure ENABLE_QUADRATIC_ADVANCE is defined .
381 L is the linear factor and seems to be working better then the quadratic
            dependency .
382 */
383 #define EXT1_ADVANCE_K 0.0f
384 #define EXT1_ADVANCE_L 0.0f
385
386 #define EXT1_WAIT_RETRACT_TEMP 150
387 #define EXT1_WAIT_RETRACT_UNITS 40

```

```

388 #define EXT1_SELECT_COMMANDS "M120 S5 P15\nM117 Extruder 2"
389 #define EXT1_DESELECT_COMMANDS ""
390 /** The extruder cooler is a fan to cool the extruder when it is heating. If you
   turn the extruder on, the fan goes on. */
391 #define EXT1_EXTRUDER_COOLER_PIN -1
392 /** PWM speed for the cooler fan. 0=off 255=full speed */
393 #define EXT1_EXTRUDER_COOLER_SPEED 255
394
395 /** If enabled you can select the distance your filament gets retracted during a
396 M140 command, after a given temperature is reached. */
397 #define RETRACT_DURING_HEATUP true
398
399 /** PID control only works target temperature +/- PID_CONTROL_RANGE.
400 If you get much overshoot at the first temperature set, because the heater is
   going full power to long, you
401 need to increase this value. For one 6.8 Ohm heater 10 is ok. With two 6.8 Ohm
   heater use 15.
402 */
403 #define PID_CONTROL_RANGE 15
404
405 /** Skip wait, if the extruder temperature is already within x degrees. Only
   fixed numbers, 0 = off */
406 #define SKIP_M109_IF_WITHIN 2
407
408 /** | brief Set PID scaling
409
410 PID values assume a usable range from 0-255. This can be further limited to
   EXT0_PID_MAX by to methods.
411 Set the value to 0: Normal computation, just clip output to EXT0_PID_MAX if
   computed value is too high.
412 Set value to 1: Scale PID by EXT0_PID_MAX/256 and then clip to EXT0_PID_MAX.
413 If your EXT0_PID_MAX is low, you should prefer the second method.
414 */
415 #define SCALE_PID_TO_MAX 0
416
417 /** Temperature range for target temperature to hold in M109 command. 5 means
   +/-5°C
418
419 Uncomment define to use force the temperature into the range for given
   watchperiod.
420 */
421 // #define TEMP_HYSTESIS 5
422
423 /** Userdefined thermistor table
424
425 There are many different thermistors, which can be combined with different
   resistors. This result
426 in unpredictable number of tables. As a resolution, the user can define one
   table here, that can
427 be used as type 5 for thermister type in extruder/heated bed definition. Make
   sure, the number of entries
428 matches the value in NUM_TEMPS_USERTHERMISTOR0. If you span definition over
   multiple lines, make sure to end
429 each line, except the last, with a backslash. The table format is {{adc1,temp1
   },{adc2,temp2}...} with
430 increasing adc values. For more informations, read
431 http://hydraraptor.blogspot.com/2007/10/measuring-temperature-easy-way.html
432
433 If you have a sprinter temperature table, you have to multiply the first value
   with 4 and the second with 8.
434 This firmware works with increased precision, so the value reads go from 0 to
   4095 and the temperature is
435 temperature*8.
436

```

```

437 If you have a PTC thermistor instead of a NTC thermistor, keep the adc values
     increasing and use the thermistor types 50-52 instead of 5-7!
438 */
439 /** Number of entries in the user thermistortable 0. Set to 0 to disable it. */
440
441 // ***** MM2 Heated bed 15k-ohm thermistor
442 // ***** Thermistor lookup table for RepRap Temperature Sensor Boards (http://make.rrrf.org/ts)
443 // Made with createTemperatureLookup.py (http://svn.reprap.org/trunk/reprap/firmware/Arduino/utilities/createTemperatureLookup.py)
444 // ./createTemperatureLookup.py --r0=15000 --t0=25 --r1=0 --r2=4700 --beta=3528
     --max-adc=1023
445 // r0: 15000
446 // t0: 25
447 // r1: 0
448 // r2: 4700
449 // beta: 3528
450 // max adc: 1023
451
452 #define NUM_TEMPS_USERTHERMISTOR0 20
453 #define USER_THERMISTORTABLE0 {\
454     {1*4, 669*8}, \
455     {54*4, 180*8}, \
456     {107*4, 140*8}, \
457     {160*4, 119*8}, \
458     {213*4, 104*8}, \
459     {266*4, 93*8}, \
460     {319*4, 83*8}, \
461     {372*4, 75*8}, \
462     {425*4, 68*8}, \
463     {478*4, 61*8}, \
464     {531*4, 55*8}, \
465     {584*4, 48*8}, \
466     {637*4, 42*8}, \
467     {690*4, 36*8}, \
468     {743*4, 29*8}, \
469     {796*4, 22*8}, \
470     {849*4, 14*8}, \
471     {902*4, 5*8}, \
472     {955*4, -7*8}, \
473     {1008*4, -35*8}}}
474
475
476 // ***** MM2 AddTherm 100k-ohm thermistor (generated from
     BETA value) *****
477 /** Number of entries in the user thermistortable 1. Set to 0 to disable it. */
478 // ./createTemperatureLookup.py --r0=100000 --t0=25 --r1=0 --r2=4700 --beta=3977
     --max-adc=1023
479 // r0: 100000
480 // t0: 25
481 // r1: 0
482 // r2: 4700
483 // beta: 3977
484 // max adc: 1023
485
486 /*
487 #define NUM_TEMPS_USERTHERMISTOR1 20
488 #define USER_THERMISTORTABLE1 {\
489     {1*4, 913*8}, |
490     {54*4, 264*8}, |
491     {107*4, 215*8}, |
492     {160*4, 189*8}, |
493     {213*4, 171*8}, |
494     {266*4, 157*8}, |

```

```

495 {319*4, 145*8}, |
496 {372*4, 135*8}, |
497 {425*4, 127*8}, |
498 {478*4, 118*8}, |
499 {531*4, 110*8}, |
500 {584*4, 103*8}, |
501 {637*4, 95*8}, |
502 {690*4, 88*8}, |
503 {743*4, 80*8}, |
504 {796*4, 71*8}, |
505 {849*4, 62*8}, |
506 {902*4, 50*8}, |
507 {955*4, 34*8}, |
508 {1008*4, 2*8} }

509 */
510
511 // **** AddTherm 100k-ohm thermistor (thermistor
512 batch #2) ****
513 /** Number of entries in the user thermistortable 1. Set to 0 to disable it. */
514 #define NUM_TEMPS_USERTHERMISTOR1 34
515 #define USER_THERMISTORTABLE1 {\
516 {0, 4000}, \
517 {92, 2399}, \
518 {105, 2337}, \
519 {121, 2271}, \
520 {140, 2203}, \
521 {162, 2135}, \
522 {189, 2063}, \
523 {51*4, 235.51*8}, \
524 {51*4, 235.51*8}, \
525 {55*4, 231.45*8}, \
526 {61*4, 225.88*8}, \
527 {64*4, 223.30*8}, \
528 {72*4, 216.96*8}, \
529 {77*4, 213.35*8}, \
530 {84*4, 208.67*8}, \
531 {91*4, 204.37*8}, \
532 {109*4, 194.66*8}, \
533 {129*4, 185.60*8}, \
534 {155*4, 175.72*8}, \
535 {169*4, 171.07*8}, \
536 {184*4, 166.50*8}, \
537 {202*4, 161.48*8}, \
538 {212*4, 158.88*8}, \
539 {264*4, 147.08*8}, \
540 {1771, 960}, \
541 {2357, 800}, \
542 {2943, 640}, \
543 {3429, 480}, \
544 {3760, 320}, \
545 {3869, 240}, \
546 {3912, 200}, \
547 {3948, 160}, \
548 {4077, -160}, \
549 {4094, -440}};

550
551 /* ===== Addterm 100k thermistor (thermistor batch #3) =====
552 */
553 /** Number of entries in the user thermistortable 2. Set to 0 to disable it. */
554 #define NUM_TEMPS_USERTHERMISTOR2 28
555 #define USER_THERMISTORTABLE2 {\
556 {0, 4000}, \
557 {92, 2399}, \
558 {105, 2337}, \

```

```

558 {121, 2271}, \
559 {140, 2203}, \
560 {162, 2135}, \
561 {189, 2063}, \
562 {222, 1988}, \
563 {261, 1913}, \
564 {308, 1836}, \
565 {365, 1757}, \
566 {434, 1676}, \
567 {519, 1592}, \
568 {621, 1509}, \
569 {744, 1425}, \
570 {891, 1341}, \
571 {1067, 1257}, \
572 {1272, 1175}, \
573 {1771, 960}, \
574 {2357, 800}, \
575 {2943, 640}, \
576 {3429, 480}, \
577 {3760, 320}, \
578 {3869, 240}, \
579 {3912, 200}, \
580 {3948, 160}, \
581 {4077, -160}, \
582 {4094, -440}};
583
584 /** If defined, creates a thermistortable at startup.
585
586 If you dont feel like computing the table on your own, you can use this generic
method. It is
587 a simple approximation which may be not as accurate as a good table computed
from the reference
588 values in the datasheet. You can increase precision if you use a temperature/
resistance for
589 R0/T0, which is near your operating temperature. This will reduce precision for
lower temperatures,
590 which are not realy important. The resistors must fit the following schematic:
591 @code
592 VREF —— R2 ——+—— Termistor ——+—— GND
593           |           |
594           +—— R1 ——+
595           |           |
596           +—— Capacitor —+
597           |
598           V measured
599 @endcode
600
601 If you don't have R1, set it to 0.
602 The capacitor is for reducing noise from long thermistor cable. If you don't
have have one, it's OK.
603
604 If you don't need the generic table, uncomment the following define.
605 */
606 #define USE_GENERIC_THERMISTORTABLE_1
607
608 /* Some examples for different thermistors:
609
610 EPCOS B57560G104+ : R0 = 100000 T0 = 25 Beta = 4036
611 EPCOS 100K Thermistor (B57560G1104F) : R0 = 100000 T0 = 25 Beta = 4092
612 ATC Semitec 104GT-2 : R0 = 100000 T0 = 25 Beta = 4267
613 Honeywell 100K Thermistor (135-104LAG-J01) : R0 = 100000 T0 = 25 Beta = 3974
614
615 */
616
617 /** Reference resistance */

```

```

618 #define GENERIC_THERM1_R0 100000
619 /** Temperature at reference resistance */
620 #define GENERIC_THERM1_T0 25
621 /** Beta value of thermistor
622
623 You can use the beta from the datasheet or compute it yourself. See
624 http://reprap.org/wiki/MeasuringThermistorBeta
625 for more details.
626 */
627 #define GENERIC_THERM1_BETA 3977
628 /** Start temperature for generated thermistor table */
629 #define GENERIC_THERM1_MIN_TEMP -10
630 /** End Temperature for generated thermistor table */
631 #define GENERIC_THERM1_MAX_TEMP 300
632 #define GENERIC_THERM1_R1 0
633 #define GENERIC_THERM1_R2 4700
634
635 // The same for table 2 and 3 if needed
636
637 #define USE_GENERIC_THERMISTORTABLE_2
638 #define GENERIC_THERM2_R0 15000
639 #define GENERIC_THERM2_T0 25
640 #define GENERIC_THERM2_BETA 3528
641 #define GENERIC_THERM2_MIN_TEMP 20
642 #define GENERIC_THERM2_MAX_TEMP 150
643 #define GENERIC_THERM2_R1 0
644 #define GENERIC_THERM2_R2 4700
645
646 // #define USE_GENERIC_THERMISTORTABLE_3
647 #define GENERIC_THERM3_R0 1042.7
648 #define GENERIC_THERM3_T0 170
649 #define GENERIC_THERM3_BETA 4036
650 #define GENERIC_THERM3_MIN_TEMP -20
651 #define GENERIC_THERM3_MAX_TEMP 300
652 #define GENERIC_THERM3_R1 0
653 #define GENERIC_THERM3_R2 4700
654
655 /** Supply voltage to ADC, can be changed by setting ANALOG_REF below to
       different value. */
656 #define GENERIC_THERM_VREF 5
657 /** Number of entries in generated table. One entry takes 4 bytes. Higher number
       of entries increase computation time too.
658 Value is used for all generic tables created. */
659 #define GENERIC_THERM_NUM_ENTRIES 33
660
661 // uncomment the following line for MAX6675 support.
662 // #define SUPPORT_MAX6675
663
664 // ##### Heated bed configuration #####
665
666 /** | brief Set true if you have a heated bed connected to your board, false if
       not */
667 #define HAVE_HEATED_BED true
668
669 #define HEATED_BED_MAX_TEMP 125
670 /** Skip M190 wait, if heated bed is already within x degrees. Fixed numbers
       only, 0 = off. */
671 #define SKIP_M190_IF_WITHIN 3
672
673 // Select type of your heated bed. It's the same as for EXT0_TEMPSENSOR_TYPE
674 // set to 0 if you don't have a heated bed
675 #define HEATED_BED_SENSOR_TYPE 5
676 /** Analog pin of analog sensor to read temperature of heated bed. */
677 #define HEATED_BED_SENSOR_PIN TEMP_BED_PIN
678 /** | brief Pin to enable heater for bed. */

```

```

679 #define HEATED_BED_HEATER_PIN HEATER_BED_PIN
680 // How often the temperature of the heated bed is set (msec)
681 #define HEATED_BED_SET_INTERVAL 5000
682
683 /**
684 Heat manager for heated bed:
685 0 = Bang Bang, fast update
686 1 = PID controlled
687 2 = Bang Bang, limited check every HEATED_BED_SET_INTERVAL. Use this with relais
       driven beds to save life
688 */
689 #define HEATED_BED_HEAT_MANAGER 1
690 /** | brief The maximum value, I-gain can contribute to the output.
691
692 A good value is slightly higher then the output needed for your temperature.
693 Values for starts:
694 130 => PLA for temperatures from 170–180°C
695 180 => ABS for temperatures around 240°C
696
697 The precise values may differ for different nozzle/resistor combination.
698 Overridden if EEPROM activated.
699 */
700 #define HEATED_BED_PID_INTEGRAL_DRIVE_MAX 255
701 /** | brief lower value for integral part
702
703 The I state should converge to the exact heater output needed for the target
       temperature.
704 To prevent a long deviation from the target zone, this value limits the lower
       value.
705 A good start is 30 lower then the optimal value. You need to leave room for
       cooling.
706 Overridden if EEPROM activated.
707 */
708 #define HEATED_BED_PID_INTEGRAL_DRIVE_MIN 80
709 /** P-gain. Overridden if EEPROM activated. */
710 #define HEATED_BED_PID_PGAİN 15
711 /** I-gain Overridden if EEPROM activated.*/
712 #define HEATED_BED_PID_IGAIN 0.9
713 /** Dgain. Overridden if EEPROM activated.*/
714 #define HEATED_BED_PID_DGAIN 40
715 // maximum time the heater is can be switched on. Max = 255. Overridden if
       EEPROM activated.
716 #define HEATED_BED_PID_MAX 255
717
718 /** Include PID control for all heaters. */
719 #define TEMP_PID true
720
721 //// Experimental watchdog and minimal temp
722 // The watchdog waits for the watchperiod in milliseconds whenever an M104 or
       M109 increases the target temperature
723 // If the temperature has not increased at the end of that period, the target
       temperature is set to zero. It can be reset with another M104/M109
724 // #define WATCHPERIOD 5000 //5 seconds
725
726 //// The minimal temperature defines the temperature below which the heater will
       not be enabled
727 #define MINTEMP 5
728
729 //// Experimental max temp
730 // When temperature exceeds max temp, your heater will be switched off.
731 // This feature exists to protect your hotend from overheating accidentally, but
       *NOT* from thermistor short/failure!
732 // You should use MINTEMP for thermistor short/failure protection.
733 #define MAXTEMP 240
734

```

```

735 /** | brief Used reference , normally ANALOG_REF_AVCC or ANALOG_REF_AREF for
    experts ANALOG_REF_INT_2_56 = 2.56V and ANALOG_REF_INT_1_1=1.1V internally
    generated */
736 #define ANALOG_REF ANALOG_REF_AVCC
737
738
739 // ##### Endstop configuration #####
740 // ## Endstop configuration ##
741 // #####
742
743 /* By default all endstops are pulled up to high. You need a pullup if you
744 use a mechanical endstop connected with gnd. Set value to false for no pullup
745 on this endstop.
746 */
747 #define ENDSTOP_PULLUP_X_MIN true
748 #define ENDSTOP_PULLUP_Y_MIN true
749 #define ENDSTOP_PULLUP_Z_MIN true
750 #define ENDSTOP_PULLUP_X_MAX true
751 #define ENDSTOP_PULLUP_Y_MAX true
752 #define ENDSTOP_PULLUP_Z_MAX true
753
754 //set to true to invert the logic of the endstops
755 #define ENDSTOP_X_MIN_INVERTING false
756 #define ENDSTOP_Y_MIN_INVERTING false
757 #define ENDSTOP_Z_MIN_INVERTING false
758 #define ENDSTOP_X_MAX_INVERTING false
759 #define ENDSTOP_Y_MAX_INVERTING false
760 #define ENDSTOP_Z_MAX_INVERTING false
761
762 // Set the values true where you have a hardware endstop. The Pin number is
    taken from pins.h.
763
764 #define MIN_HARDWARE_ENDSTOP_X true
765 #define MIN_HARDWARE_ENDSTOP_Y true
766 #define MIN_HARDWARE_ENDSTOP_Z true
767 #define MAX_HARDWARE_ENDSTOP_X true
768 #define MAX_HARDWARE_ENDSTOP_Y true
769 #define MAX_HARDWARE_ENDSTOP_Z true
770
771 //If your axes are only moving in one direction , make sure the endstops are
    connected properly.
772 //If your axes move in one direction ONLY when the endstops are triggered , set
    ENDSTOPS_INVERTING to true here
773
774
775
776 //// ADVANCED SETTINGS – to tweak parameters
777
778 // For Inverting Stepper Enable Pins (Active Low) use 0 , Non Inverting (Active
    High) use 1
779 #define X_ENABLE_ON 0
780 #define Y_ENABLE_ON 0
781 #define Z_ENABLE_ON 0
782
783 // Disables axis when it's not being used.
784 #define DISABLE_X false
785 #define DISABLE_Y false
786 #define DISABLE_Z false
787 #define DISABLE_E false
788
789 // Inverting axis direction
790 #define INVERT_X_DIR false
791 #define INVERT_Y_DIR false
792 #define INVERT_Z_DIR false
793

```

```

794 ///////////////////////////////////////////////////////////////////
795 // Sets direction of endstops when homing; 1=MAX, -1=MIN
796 #define X_HOME_DIR -1
797 #define Y_HOME_DIR -1
798 #define Z_HOME_DIR 1
799
800 // Delta robot radius endstop
801 #define max_software_endstop_r true
802
803 // If true, axis won't move to coordinates less than zero.
804 #define min_software_endstop_x true
805 #define min_software_endstop_y true
806 #define min_software_endstop_z false
807
808 // If true, axis won't move to coordinates greater than the defined lengths below
809
810 #define max_software_endstop_x true
811 #define max_software_endstop_y true
812 #define max_software_endstop_z true
813
814 // If during homing the endstop is reached, how many mm should the printer move
815 // back for the second try
816 #define ENDSTOP_X_BACK_MOVE 5
817 #define ENDSTOP_Y_BACK_MOVE 5
818 #define ENDSTOP_Z_BACK_MOVE 2
819
820 // For higher precision you can reduce the speed for the second test on the
821 // endstop
822 // during homing operation. The homing speed is divided by the value. 1 = same
823 // speed, 2 = half speed
824 #define ENDSTOP_X_RETEST_REDUCTION_FACTOR 2
825 #define ENDSTOP_Y_RETEST_REDUCTION_FACTOR 2
826 #define ENDSTOP_Z_RETEST_REDUCTION_FACTOR 2
827
828 // When you have several endstops in one circuit you need to disable it after
829 // homing by moving a
830 // small amount back. This is also the case with H-belt systems.
831 #define ENDSTOP_X_BACK_ON_HOME 0
832 #define ENDSTOP_Y_BACK_ON_HOME 0
833 #define ENDSTOP_Z_BACK_ON_HOME 0
834
835 // You can disable endstop checking for print moves. This is needed, if you get
836 // sometimes
837 // false signals from your endstops. If your endstops don't give false signals,
838 // you
839 // can set it on for safety.
840 #define ALWAYS_CHECK_ENDSTOPS false
841
842 // maximum positions in mm - only fixed numbers!
843 // For delta robot Z_MAX_LENGTH is maximum travel of the towers and should be
844 // set to the distance between the hotend
845 // and the platform when the printer is at its home position.
846 // If EEPROM is enabled these values will be overridden with the values in the
847 // EEPROM
848
849 #define X_MAX_LENGTH 200
850 #define Y_MAX_LENGTH 200
851 #define Z_MAX_LENGTH 150
852
853 // Coordinates for the minimum axis. Can also be negative if you want to have
854 // the bed start at 0 and the printer can go to the left side
855 // of the bed. Maximum coordinate is given by adding the above X_MAX_LENGTH
856 // values.
857 #define X_MIN_POS 0
858 #define Y_MIN_POS 0
859 #define Z_MIN_POS 0

```

```

848
849 // ##### Movement settings #####
850 // ## Movement settings ##
851 // #####
852
853 // Microstep setting (Only functional when stepper driver microstep pins are
     connected to MCU. Currently only works for RAMBO boards)
854 #define MICROSTEP_MODES {16,16,16,16,16} // [1,2,4,8,16]
855
856 // Motor Current setting (Only functional when motor driver current ref pins are
     connected to a digital trimpot on supported boards)
857 // {X, Y, Z, E1, E2}
858 // Vref = Imax * (8 * Rs)
859 // W = Vref/1.66 * 255
860
861 // RAMBo board: Rs = 0.1 ohm
862 // Values 0-255 (RAMBo 135 = ~0.75A, 185 = ~1A, 222 = ~1.2A)
863 #define MOTOR_CURRENT {100,100,135,110,110}
864
865 // 3D Master board
866 // #define MOTOR_CURRENT {35713,35713,35713,35713,35713} // Values 0-65535 (3D
     Master 35713 = ~1A)
867
868
869 #define STEP_COUNTER
870
871 // Delta settings
872 #if DRIVE_SYSTEM==3
873 /** | brief Delta rod length
874 */
875 #define DELTA_DIAGONAL_ROD 250.0 // mm
876
877 /** | brief Number of segments to generate for delta conversions per second of
     move
878 */
879 #define DELTA_SEGMENTS_PER_SECOND_PRINT 200 // Move accurate setting for print
     moves
880 #define DELTA_SEGMENTS_PER_SECOND_MOVE 70 // Less accurate setting for other
     moves
881
882 /** | brief Horizontal offset of the universal joints on the end effector (moving
     platform).
883 */
884 #define END_EFFECTOR_HORIZONTAL_OFFSET 33
885
886 /** | brief Horizontal offset of the universal joints on the vertical carriages.
887 */
888 #define CARRIAGE_HORIZONTAL_OFFSET 18
889
890 /** | brief Printer radius in mm, measured from the center of the print area to
     the vertical smooth rod.
891 */
892 #define PRINTER_RADIUS 175
893
894 /** | brief Horizontal distance bridged by the diagonal push rod when the end
     effector is in the center. It is pretty close to 50% of the push rod length
     (250 mm).
895 */
896 #define DELTA_RADIUS (PRINTER_RADIUS-END_EFFECTOR_HORIZONTAL_OFFSET-
     CARRIAGE_HORIZONTAL_OFFSET)
897
898 /** | brief Enable counter to count steps for Z max calculations
899 */
900 #define STEP_COUNTER
901

```

```

902 /** | brief Experimental calibration utility for delta printers
903 */
904 #define SOFTWARE_LEVELING
905
906 #endif
907
908 /** After x seconds of inactivity, the stepper motors are disabled.
909 Set to 0 to leave them enabled.
910 This helps cooling the Stepper motors between two print jobs.
911 Overridden if EEPROM activated.
912 */
913 #define STEPPER_INACTIVE_TIME 30L
914 /** After x seconds of inactivity, the system will go down as far it can.
915 It will at least disable all stepper motors and heaters. If the board has
916 a power pin, it will be disabled, too.
917 Set value to 0 for disabled.
918 Overridden if EEPROM activated.
919 */
920 #define MAX_INACTIVE_TIME 300L
921 /** Maximum feedrate, the system allows. Higher feedrates are reduced to these
922 values.
923 The axis order in all axis related arrays is X, Y, Z
924 Overridden if EEPROM activated.
925 */
926 #define MAX_FEEDRATE_X 500
927 #define MAX_FEEDRATE_Y 400
928 #define MAX_FEEDRATE_Z 5
929 /** Speed in mm/s for finding the home position. Overridden if EEPROM activated
930 */
931 #define HOMING_FEEDRATE_X 50
932 #define HOMING_FEEDRATE_Y 50
933 #define HOMING_FEEDRATE_Z 5
934 /* If you have a backlash in both z-directions, you can use this. For most
935 printer, the bed will be pushed down by it's
936 own weight, so this is nearly never needed. */
937 #define ENABLE_BACKLASH_COMPENSATION true
938 #define Z_BACKLASH 0
939 #define X_BACKLASH 0
940 #define Y_BACKLASH 0
941 /** Comment this to disable ramp acceleration */
942 #define RAMP_ACCELERATION 1
943
944 /** If your stepper needs a longer high signal then given, you can add a delay
945 here.
946 The delay is realized as a simple loop wasting time, which is not available for
947 other
948 computations. So make it as low as possible. For the most common drivers no
949 included delay is already enough.
950 */
951 #define STEPPER_HIGH_DELAY 0
952
953 /** The firmware can only handle 16000Hz interrupt frequency cleanly. If you
954 need higher speeds
955 a faster solution is needed, and this is to double/quadruple the steps in one
956 interrupt call.
957 This is like reducing your 1/16th microstepping to 1/8 or 1/4. It is much
958 cheaper then 1 or 3
959 additional stepper interrupts with all it's overhead. As a result you can go as
960 high as
961 40000Hz.
962 */

```

```

957 #define STEP_DOUBLER_FREQUENCY 12000
958 /** If you need frequencies off more then 30000 you definitely need to enable
   this. If you have only 1/8 stepping
959 enabling this may cause to stall your moves when 20000Hz is reached.
960 */
961 #define ALLOW_QUADSTEPPING true
962 /** If you reach STEP_DOUBLER_FREQUENCY the firmware will do 2 or 4 steps with
   nearly no delay. That can be too fast
963 for some printers causing an early stall.
964
965 */
966 #define DOUBLE_STEP_DELAY 1 // time in us
967
968 /** The firmware supports trajectory smoothing. To achieve this, it divides the
   stepsize by 2, resulting in
969 the double computation cost. For slow movements this is not an issue, but for
   really fast moves this is
970 too much. The value specified here is the number of clock cycles between a step
   on the driving axis.
971 If the interval at full speed is below this value, smoothing is disabled for
   that line.*/
972 #define MAX_HALFSTEP_INTERVAL 1999
973
974 //// Acceleration settings
975
976 /** | brief X, Y, Z max acceleration in mm/s^2 for printing moves or retracts.
   Make sure your printer can go that high!
977 Overridden if EEPROM activated.
978 */
979 #define MAX_ACCELERATION_UNITS_PER_SQ_SECOND_X 1500
980 #define MAX_ACCELERATION_UNITS_PER_SQ_SECOND_Y 1500
981 #define MAX_ACCELERATION_UNITS_PER_SQ_SECOND_Z 100
982
983 /** | brief X, Y, Z max acceleration in mm/s^2 for travel moves. Overridden if
   EEPROM activated.*/
984 #define MAX_TRAVEL_ACCELERATION_UNITS_PER_SQ_SECOND_X 3000
985 #define MAX_TRAVEL_ACCELERATION_UNITS_PER_SQ_SECOND_Y 3000
986 #define MAX_TRAVEL_ACCELERATION_UNITS_PER_SQ_SECOND_Z 100
987
988 /** | brief Maximum allowable jerk.
989
990 Caution: This is no real jerk in a physical meaning.
991
992 The jerk determines your start speed and the maximum speed at the join of two
   segments.
993 It's unit is mm/s. If the printer is standing still, the start speed is jerk/2.
   At the
994 join of two segments, the speed difference is limited to the jerk value.
995
996 Examples:
997 For all examples jerk is assumed as 40.
998
999 Segment 1: vx = 50, vy = 0
1000 Segment 2: vx = 0, vy = 50
1001 v_diff = sqrt((50-0)^2+(0-50)^2) = 70.71
1002 v_diff > jerk => vx_1 = vy_2 = jerk/v_diff*vx_1 = 40/70.71*50 = 28.3 mm/s at the
   join
1003
1004 Segment 1: vx = 50, vy = 0
1005 Segment 2: vx = 35.36, vy = 35.36
1006 v_diff = sqrt((50-35.36)^2+(0-35.36)^2) = 38.27 < jerk
1007 Corner can be printed with full speed of 50 mm/s
1008
1009 Overridden if EEPROM activated.
1010 */

```

```

1011 #define MAX_JERK 20.0
1012 #define MAX_ZJERK 0.3
1013
1014 /** | brief Number of moves we can cache in advance.
1015
1016 This number of moves can be cached in advance. If you wan't to cache more,
   increase this. Especially on
1017 many very short moves the cache may go empty. The minimum value is 5.
1018 */
1019 #define MOVE_CACHE_SIZE 16
1020
1021 /** | brief Low filled cache size.
1022
1023 If the cache contains less then MOVE_CACHE_LOW segments, the time per segment is
   limited to LOW_TICKS_PER_MOVE clock cycles.
1024 If a move would be shorter, the feedrate will be reduced. This should prevent
   buffer underflows. Set this to 0 if you
1025 don't care about empty buffers during print.
1026 */
1027 #define MOVE_CACHE_LOW 10
1028 /** | brief Cycles per move, if move cache is low.
1029
1030 This value must be high enough, that the buffer has time to fill up. The problem
   only occurs at the beginning of a print or
1031 if you are printing many very short segments at high speed. Higher delays here
   allow higher values in PATH_PLANNER_CHECK_SEGMENTS.
1032 */
1033 #define LOW_TICKS_PER_MOVE 250000
1034
1035 // ##### Extruder control #####
1036 // ## Extruder control ##
1037 // #####
1038
1039 /** | brief Prescale factor, timer0 runs at.
1040
1041 All known arduino boards use 64. This value is needed for the extruder timing.
   */
1042 #define TIMER0_PRESCALE 64
1043
1044 /* | brief Minimum temperature for extruder operation
1045
1046 This is a saftey value. If your extruder temperature is below this temperature,
   no
1047 extruder steps are executed. This is to prevent your extruder to move unless the
   filament
1048 is at least molten. After havong some complains that the extruder does not work,
   I leave
1049 it 0 as default.
1050 */
1051
1052 #define MIN_EXTRUDER_TEMP 0
1053 /** | brief Activate ooze prevention system
1054
1055 The ooze prevention system tries to prevent ooze, by a fast retract of the
   filament every time
1056 printing stops. Most slicing software have already an option to do this. Using
   OPS_MODE=1 will
1057 in fact mimic this. This works good, but can increase printing time. To reduce
   the additional
1058 waiting time, the OPS has a fast mode, which performs the retraction during the
   travelling move.
1059 The only reason, your slicer doesn't do it, is because it can't tell. There is
   simple no
1060 G-Code command telling the firmware to do that.
1061

```

```

1062 You can always compile including OPS. Then you can disable/enable it anytime you
      want. To disable it
1063 set USE_OPS 0
1064
1065 Caution: Don't enable anti-ooze in your slicer if you are using this.
1066 */
1067 #define USE_OPS 1
1068
1069 /** | brief Sets the ops operation mode
1070
1071 0: Off
1072 1: Classic mode. Stop head, retract move to target, push filament back.
1073 2: Fast mode. Retract during move, start pushing back the filament during move.
      For safty, we start
1074 at with a low speed and wait for the push back, before the printmove starts.
      Normally there is some
1075 time needed to wait for the filament.
1076
1077 Overridden if EEPROM activated.
1078 */
1079 #define OPS_MODE 0
1080
1081 /** | brief Minimum distance for retraction.
1082
1083 If a travel move is shorter than this distance, no retraction will occur. This
      is to prevent
1084 retraction with infill, where the angle to the perimeter needs a short stop.
      Unit is mm.
1085 Overridden if EEPROM activated.
1086 */
1087 #define OPS_MIN_DISTANCE 0.8
1088
1089 /** | brief Move printhead only after x% of retract distance have been retracted.
1090
1091 Overridden if EEPROM activated.*/
1092 #define OPS_MOVE_AFTER 50.0
1093 /** | brief Retraction distance in mm. If you want to enable OPS only sometimes,
      compile with
1094 OPS support and set retraction distance to 0. If you set it to e.g. 3 in your
      eeprom settings it is enabled.
1095 Overridden if EEPROM activated.*/
1096 #define OPS_RETRACT_DISTANCE 1.5
1097
1098 /** | brief Backslash produced by extruder reversal
1099
1100 If you are using a bowden extruder, you may need some extra distance to push the
      filament back into the
1101 original place. This is the value you enter here. Unit is mm.
1102 Overridden if EEPROM activated.
1103 */
1104 #define OPS_RETRACT_BACKLASH 0.0
1105
1106 /** | brief Enable advance algorithm.
1107
1108 Without a correct adjusted advance algorithm, you get blobs at points, where
      acceleration changes. The
1109 effect increases with speed and acceleration difference. Using the advance
      method decreases this effect.
1110 For more informations, read the wiki.
1111 */
1112 #define USE_ADVANCE
1113
1114 /** | brief enables quadratic component.
1115

```

```

1116 Uncomment to allow a quadratic advance dependency. Linear is the dominant value,
       so no real need
1117 to activate the quadratic term. Only adds lots of computations and storage usage
       . */
1118 // #define ENABLE_QUADRATIC_ADVANCE
1119
1120
1121 // ##### Communication configuration #####
1122 // ## Communication configuration ##
1123 // #####
1124
1125 //// AD595 THERMOCOUPLE SUPPORT UNTESTED... USE WITH CAUTION!!!!
1126
1127 /** | brief Communication speed.
1128
1129 - 250000 : Fastest with errorrate of 0% with 16 or 32 MHz - update wiring_serial.
       c in your board files. See boards/readme.txt
1130 - 115200 : Fast, but may produce communication errors on quite regular basis,
       Error rate ~3,5%
1131 - 76800 : Best setting for Arduino with 16 MHz, Error rate 0,2% page 198 AVR1284
       Manual. Result: Faster communication than 115200
1132 - 57600 : Should produce nearly no errors, on my gen 6 it's faster than 115200
       because there are no errors slowing down the connection
1133 - 38600
1134
1135 Overridden if EEPROM activated.
1136 */
1137 // #define BAUDRATE 76800
1138 // #define BAUDRATE 57600
1139 #define BAUDRATE 115200
1140 // #define BAUDRATE 250000
1141
1142 /**
1143 Some boards like Gen7 have a power on pin, to enable the atx power supply. If
       this is defined,
1144 the power will be turned on without the need to call M80 if initially started.
1145 */
1146 #define ENABLE_POWER_ON_STARTUP
1147
1148 /** What shall the printer do, when it receives an M112 emergency stop signal?
1149 0 = Disable heaters/motors, wait for ever until someone presses reset.
1150 1 = restart by resetting the AVR controller. The USB connection will not reset
       if managed by a different chip!
1151 */
1152 #define KILL_METHOD 1
1153
1154 /** | brief Cache size for incoming commands.
1155
1156 There should be no reason to increase this cache. Commands are nearly
       immediately send to
1157 execution.
1158 */
1159 #define GCODE_BUFFER_SIZE 2
1160 /** Appends the linenumber after every ok send, to acknowledge the received
       command. Uncomment for plain ok ACK if your host has problems with this */
1161 // #define ACK_WITH_LINENUMBER
1162 /** Communication errors can swallow part of the ok, which tells the host
       software to send
1163 the next command. Not receiving it will cause your printer to stop. Sending this
       string every
1164 second, if our queue is empty should prevent this. Uncomment if you don't want
       this feature. */
1165 #define WAITING_IDENTIFIER "wait"
1166
1167 /** | brief Sets time for echo debug

```

```

1168
1169 You can set M111 1 which enables ECHO of commands send. This define specifies
1170 the position ,
1170 when it will be executed. In the original FiveD software , echo is done after
1171 receiving the
1171 command. With checksum you know, how it looks from the sending string . With this
1172 define
1172 uncommented, you will see the last command executed. To be more specific: It is
1173 written after
1173 execution. This helps tracking errors , because there may be 8 or more commands
1174 in the queue
1174 and it is otherwise difficult to know, what your reprap is currently doing .
1175 */
1176 #define ECHO_ON_EXECUTE
1177
1178 /** | brief EEPROM storage mode
1179
1180 Set the EEPROM_MODE to 0 if you always wan't to use the settings in this
1180 configuration file. If not ,
1181 set it to a value not stored in the first EEPROM-byte used. If you later want to
1181 overwrite your current
1182 eeprom settings with configuration defaults , just select an other value. On the
1182 first call to epr_init()
1183 it will detect a mismatch of the first byte and copys default values into EEPROM
1183 . If the first byte
1184 matches , the stored values are used to overwrite the settings .
1185
1186 IMPORTANT: With mode <>0 some changes in configuration.h are not set any more ,
1186 as they are
1187 taken from the EEPROM.
1188 */
1189 #define EEPROM_MODE 3
1190 /** Set to false to disable SD support: */
1191 #ifndef SDSUPPORT // Some boards have sd support on board. These define the
1191 values already in pins.h
1192 #define SDSUPPORT false
1193 /** If set to false all files with longer names then 8.3 or having a tilde in
1193 the name will be hidden */
1194 #define SD_ALLOW_LONG_NAMES false
1195 // Uncomment to enable or changed card detection pin. With card detection the
1195 card is mounted on insertion .
1196 #define SDCARDDETECT -1
1197 // Change to true if you get a inserted message on removal .
1198 #define SDCARDDETECTINVERTED false
1199 #endif
1200 /** Show extended directory including file length . Don't use this with
1200 pronterface ! */
1201 #define SD_EXTENDED_DIR
1202 // If you want support for G2/G3 arc commands set to true , otherwise false .
1203 #define ARC_SUPPORT true
1204
1205 /** You can store the current position with M401 and go back to it with M402 .
1206 This works only if feature is set to true . */
1207 #define FEATURE_MEMORY_POSITION true
1208
1209 /** Should support for fan control be compiled in . If you enable this make sure
1210 the FAN pin is not the same as for your second extruder . RAMPS e.g. has FAN_PIN
1210 in 9 which
1211 is also used for the heater if you have 2 extruders connected . */
1212 #define FEATURE_FAN_CONTROL true
1213
1214 /** For displays and keys there are too many permutations to handle them all in
1214 once .
1215 For the most common available combinations you can set the controller type here ,
1215 so

```

```

1216 you don't need to configure uiconfig.h at all. Controller settings > 1 disable
1217 usage
1218 of uiconfig.h
1219 0 = no display
1220 1 = Manual definition of display and keys parameter in uiconfig.h
1221
1222 The following settings override uiconfig.h!
1223 2 = Smartcontroller on a RAMPS from reprapdiscount
1224 3 = Adafruit RGB controller
1225 4 = Folytn 3DMaster with display attached
1226 */
1227 #define FEATURE_CONTROLLER 0
1228
1229 /**
1230 Select the language to use.
1231 0 = english
1232 1 = german
1233 2 = dutch
1234 3 = brazilian portuguese
1235 */
1236 #define UI_LANGUAGE 3
1237
1238 // This is line 2 of the status display at startup
1239 #define UI_VERSION_STRING2 "Metamáquina 2"
1240
1241 /** How many ms should a single page be shown, until it is switched to the next
one.*/
1242 #define UI_PAGES_DURATION 4000
1243
1244 /** Uncomment if you don't want automatic page switching. You can still switch
the
1245 info pages with next/previous button/click-encoder */
1246 #define UI_DISABLE_AUTO_PAGESWITCH true
1247
1248 /** Time to return to info menu if x millisconds no key was pressed. Set to 0 to
disable it. */
1249 #define UI_AUTORETURN_TO_MENU_AFTER 30000
1250
1251 #define FEATURE_UI_KEYS 0
1252
1253 /* Normally cou want a next/previous actions with every click of your encoder.
1254 Unfotunately, the encoder have a different count of phase changes between clicks
.
1255 Select an encoder speed from 0 = fastest to 2 = slowest that results in one menu
move per click.
1256 */
1257 #define UI_ENCODER_SPEED 1
1258 /** | brief bounce time of keys in milliseconds */
1259 #define UI_KEY_BOUNCETIME 10
1260
1261 /** | brief First time in ms until repeat of action. */
1262 #define UI_KEY_FIRST_REPEAT 500
1263 /** | brief Reduction of repeat time until next execution. */
1264 #define UI_KEY_REDUCE_REPEAT 50
1265 /** | brief Lowest repeat time. */
1266 #define UI_KEY_MIN_REPEAT 50
1267
1268 #define FEATURE_BEEPER true
1269 /**
1270 Beeper sound definitions for short beeps during key actions
1271 and longer beeps for important actions.
1272 Parameter is is delay in microseconds and the seconds is the number of
repetitions.
1273 Values must be in range 1..255

```

```

1274 */
1275 #define BEEPER_SHORT_SEQUENCE 2,2
1276 #define BEEPER_LONG_SEQUENCE 8,8
1277
1278 // ##### Values for menu settings #####
1279 // ##                                     ##
1280 // #####                                     #####
1281
1282 // Values used for preheat
1283 #define UI_SET_PRESET_HEATED_BED_TEMP_PLA 60
1284 #define UI_SET_PRESET_EXTRUDER_TEMP_PLA 170
1285 #define UI_SET_PRESET_HEATED_BED_TEMP_ABS 110
1286 #define UI_SET_PRESET_EXTRUDER_TEMP_ABS 240
1287 // Extreme values
1288 #define UI_SET_MIN_HEATED_BED_TEMP 55
1289 #define UI_SET_MAX_HEATED_BED_TEMP 120
1290 #define UI_SET_MIN_EXTRUDER_TEMP 160
1291 #define UI_SET_MAX_EXTRUDER_TEMP 270
1292 #define UI_SET_EXTRUDER_FEEDRATE 2 // mm/sec
1293 #define UI_SET_EXTRUDER_RETRACT_DISTANCE 3 // mm
1294
1295 #endif

```

B.3 Teste de Temperaturas

```

1
2 /*
3      by Sara - sara@metamaquina.com.br
4
5      a chapa vai esquentar pra valer!
6 */
7
8 #include <math.h>
9
10 //#define MEGA
11 #define RAMBO
12
13 #ifdef MEGA
14 #define TEMP_PIN0 15
15
16
17 #endif
18
19 #ifdef RAMBO
20 #define TEMP_PIN_1 1
21 #define TEMP_PIN_2 0
22 #define TEMP_PIN_3 7
23
24
25 #endif
26
27 //float T0 = 25 +273.15; //em K
28 //float B = 4000;
29 //float R0 = 100000;
30
31 void setup(void) {
32   Serial.begin(115200);
33 //}
34
35 //float Thermistor(float RawADC) {
36   //float Temp;

```

```

37 // See http://en.wikipedia.org/wiki/Thermistor for explanation of formula
38 //float R = 1/((1/RawADC)-(1/4701));
39 //Temp = 1 / (1/T0 + 1/B * log (R/R0));
40 //Temp = Temp - 273.15; // Convert Kelvin to Celcius
41
42 //Serial.print("R: "); Serial.println(R);
43 //Serial.print("ADC: "); Serial.println(RawADC);
44 //Serial.print("T: "); Serial.println(Temp);
45 //Serial.println(" ");
46 //return Temp;
47 //}
48
49 //void printTemp(void) {
50
51 }
52
53 void loop(void) {
54     float temp1 = analogRead(TEMP_PIN_1); // Read sensor
55     float temp2 = analogRead(TEMP_PIN_2);
56     float temp3 = analogRead(TEMP_PIN_3);
57     Serial.print("ADC1: "); Serial.println(temp1);
58     Serial.print("ADC2: "); Serial.println(temp2);
59     Serial.print("ADC3: "); Serial.println(temp3);
60     Serial.println(" ");
61     delay(500);
62 }
```

B.4 Teste de Vibração

A seguir são apresentados os dois *scripts* para a realização do teste de vibração.

B.4.1 Gcode

```

1 j=1;
2 posicao = 0
3 feedrate = 10000
4 percurso = 200
5
6 for i in range(percurso):
7     posicao = posicao+1
8     print ("G1 X%d F%d" % (posicao+j, feedrate))
9     print ("G1 X%d F%d" % (posicao-j, feedrate))
10
11 posicao = 200
12
13 for i in range(percurso):
14     posicao = posicao-1
15     print ("G1 X%d F%d" % (posicao-j, feedrate))
16     print ("G1 X%d F%d" % (posicao+j, feedrate))
```

B.4.2 Printcore

```

1 import time, getopt, sys, platform, os, printcore, itertools
2 p=printcore.printcore('/dev/ttyACM0',115200)
3
4 p.connect()
5
```

```

6 time.sleep(50)
7
8 amplitude=1
9 posicao = 0
10 feedrate = 10000
11 percurso = 200
12
13 while True:
14     f=open( 'teste.gcode' )
15     for line in f.readlines():
16         time.sleep(0.05)
17         print( line )
18         p.send( line )

```

B.5 Teste de Extrusão

```

1
2 import random
3
4 extrude=5
5 retract=2
6 feedrate_e=100
7 feedrate_r=10000
8 cycle=15000
9
10 print ("M104 S185")
11 print ("G91")
12
13 for w in range(cycle):
14     print ("G1 E%d F%d" % (extrude+random.randint(0, 9), feedrate_e))
15     print ("G1 E%d F%d" % (-retract, feedrate_r))
16     print ("G4 P%d" % (random.randint(0,1000)))
17     print ("G1 E%d F%d" % (retract, feedrate_r))

```

Referências Bibliográficas

- [3DP12] 3DPrinter.net. Organovo: the bioprinting pioneer to watch. <http://www.3dprinter.net/organovo-the-bioprinting-pioneer-to-watch>, 2012. 3
- [Ata13] Anthony Atala. Ted. http://www.ted.com/talks/anthony_atala_printing_a_human_kidney.html, 2013. 3
- [Bar13a] Christopher Barnatt. 3d printing. <http://www.explainingthefuture.com/3dprinting.html>, 2013. 4
- [Bar13b] Christopher Barnatt. 3d printing directory. http://explainingthefuture.com/3d_printing_directory.html, 2013. 5, 6
- [Bar13c] Owen Barrow. 3d printing guns cody wilson. <http://owenbarrow.org/wp-admin/network/3d-printing-guns-cody-wilson>, 2013. 3
- [Bea01] Joseph J. Beaman. Solid freeform fabrication: An historical perspective. *SFF 2001 Proceedings*, 2001. 2
- [Bow13] Adrian Bowyer. Reprap. <http://reprap.org/wiki/RepRap>, 2013. 3
- [Cor95] Z Corporation. Z corporation 3d printing technology - fast, affordable and uniquely versatile. http://www.zcorp.com/documents/108_3D%20PrintingFINAL.pdf, 1995. 2
- [D.11] Contuzzi N.; Campanelli S. L.; Ludovico A. D. 3d finite element analysis in the selective laser melting process. *Int j simul model*, 2011. 2
- [Dis13] Defense Distributed. Makers of the liberator pistol. <http://defdist.org/>, 2013. 3
- [Eco13] Kor Ecologic. Urbee 2. <http://korecologic.com/>, 2013. 3
- [Fei13] Ashley Feinberg. How 3d printing gave this man his life (and face) back. <http://gizmodo.com/5993147/how-3d-printing-gave-this-man-his-life-and-face-back>, 2013. 3
- [Gri12] Todd Grimm. Layerwise builds the world's first patient specific lower jaw using laser technology. <http://www.engineering.com/3DPrinting/3DPrintingArticles/ArticleID/4253/Layerwise-Builds-the-Worlds-First-PatientSpecific-Lower-Jaw-Using-Laser-Technology.aspx>, 2012. 3
- [Hil12] Rock Hill. 3d systems- ceo and co-founder anchor list of 20 most influential people in 3d printing. <http://www.3dsystems.com/sites/www.3dsystems.com/files/06182012-3d-systems-ceo-and-co-founder-anchor-list-of-20-most-influential-people-in-3d-printing.pdf>, 2012. 2
- [Hsu13] Jeremy Hsu. 3d printing aims to deliver organs on demand. <http://www.livescience.com/39885-3d-printing-to-deliver-organs.html>, 2013. 2
- [Lay13] 3T Building Success Layer By Layer. Metal additive manufacturing (am) using direct metal laser sintering (dmls). <http://www.3trpd.co.uk/dmls.htm>, 2013. 4
- [LC09] Rui Vilar Lino Costa. Laser powder deposition. *Rapid Prototyping Journal*, 2009. 2

- [Loc13] Loctite. Technical data sheet - loctite 243. <http://tds.loctite.com/tds5/docs/243%20NEW-EN.PDF>, 2013. 28
- [Mak13] Makerbot. Makerbot thingiverse. <https://www.thingiverse.com/>, 2013. 3
- [McG88] Thomas McGee. *Principles and Methods of Temperature Measurement*. John Wiley & Sons, 1988. 24
- [Obj13] Objet. Connex500 - the polyjet matrix system. http://oldsite.objet.com/Portals/0/docs/Connex_500.pdf, 2013. 4
- [Ros09] David L. Bourell; Joseph J. Beaman Jr.; Ming C. Leu; David W. Rosen. A brief history of additive manufacturing and the 2009 roadmap for additive manufacturing: Looking back and looking ahead. *RapidTech 2009: US-TURKEY Workshop on Rapid Technologies*, 2009. 2
- [Rus13] Johnny Russell. Rambo - reprap. <http://reprap.org/wiki/Rambo>, 2013. 25
- [Sch13] Dan Schutzer. Potential impact of 3d printing on financial services. *BITS - Financial Services Roundtable*, 2013. 2
- [see13] seemecnc. SeemeCNC - 3d printers and more. https://raw.githubusercontent.com/seemeecnc/Orion_Delta/master/USER_MANUAL/ORIONUserManual.pdf, 2013. xiv, 30
- [Sha13] Shapeways. About us. <http://www.shapeways.com/about?li=footer>, 2013. 3
- [Str92] Inc. Stratasys. Fast, precise, safe prototyper with fdm, 1992. 2
- [Str08] Stratasys. Stratasys introduces abs-m30i for medical, food, and pharmaceutical equipment prototyping or production. <http://investors.stratasys.com/common/mobile/iphone/releasedetail.cfm?ReleaseID=598480&CompanyID=AMDA-FNA1K&mobileid=>, 2008. 3
- [Stu11] Peter Sturm. A historical survey of geometric computer vision. *International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2011. 2
- [tC13] Blueprinter Empowering People to Create. Affordable 3d printer with new selective heat sintering (shs) technology. <http://www.blueprinter.dk/shs.html>, 2013. 5
- [TG01] Terry Wohlers Todd Grimm. 3d printers for the home: Are they in our future? <http://www.tagrimm.com/publications/perspectives-jun2001.html>, 2001. 2
- [Tor13] BMC Toronto. Master of science in biomedical communications. http://www.bmc.med.utoronto.ca/bmcwiki/doku.php/technology:media_technologies:3dprinting, 2013. 3
- [TW12] Tim Gornet Terry Wohlers. History of additive manufacturing. *Wohlers Report*, 2012. 2
- [Woz09] Friedrich B. Prinz; Clinton L. Atwood; Richard F. Aubin; Joseph J. Beaman; Robert L. Brown; Paul S. Fussell; Allan J. Lightman; Emanuel Sachs; Lee E. Weiss; Michael J. Wozny. *Rapid Prototyping in Europe and Japan*, volume II. Perspectiva, 2009. Tradução Gilson Cesar Cardoso de Souza. 2