

2021 年度 卒 業 論 文

氷をマテリアルとした 3D プリンターの開発と改良

指導教員：羽田久一 教授

メディア学部 メディア学部 A.E.D. lab

学籍番号 M0118050

大谷真太郎

2022 年 1 月

2021 年度 卒 業 論 文 概 要				
論文題目				
氷をマテリアルとした 3D プリンターの開発と改良				
メディア学部 学籍番号 : M0118050	氏名	大谷真太郎	指導教員	羽田久一 教授
キーワード	氷、3 D プリンター			
<p>氷の造形物は彫刻で作るのが一般的だが、スキルや材料の調達などの問題で誰もが簡単に製作できる訳ではない。本稿では、先行研究である「氷をマテリアルとした 3 D プリンターの開発 著者：東京工科大学大学院 藤田大樹」の液体窒素を使い FDM 方式で氷の造形を行うプリンタを中心に問題点の改良を行う。特に問題として抱えていた、造形速度と造形精度の向上とオーバーハングの実現を目指す。その手法として、氷の粘度を変える方法と造形物のサポートの充填率のパラメーターを変えながら印刷し関係性を調査し、その結果から、氷の造形物を印刷するのに適したパラメータを発見する。これにより、ユーザーは特別な知識がなくても氷プリンター用の GCode を作ることができる。氷の造形物は、時間の経過で溶けて完全に消失する性質を持っており、これは 3D プリンターに新しい表現を与える。</p>				

目 次

1

2 第 1 章 はじめに	1
3 1.1 はじめに	1
4 1.2 現在の 3D プリンター	1
5 1.3 氷をマテリアルとした 3D プリンター	3
6 1.4 氷造形の提案	4
7 第 2 章 関連研究	5
8 2.1 関連研究	5
9 2.2 ゲルを用いて印刷する 3D プリンター [1]	5
10 2.3 Suntory-3D on the Rocks[2]	6
11 2.4 A Layered Fabric 3D Printer for Soft Inter-active Objects[3]	7
12 2.5 Additive manufacturing of optically trans-parent glass[4]	7
13 2.6 静電インクジェット式 3D プリンタによる高粘度食品材料の高精度プリント [5]	9
14 2.7 フルカラー 3D プリンター—2D 印刷から 3D 印刷へ—[6]	11
15 2.8 3D プリンタのセラミックスへの適用 [7]	12
16 2.9 3D 食用ゲルジェットプリンタによる食品創製 [8]	14
17 2.10 積彩 [9]	16
18 2.11 放電現象を利用したインクジェット型金属 3D プリンター開発に関する基礎研究 [10]	16
19 2.12 Elsa:氷を素材とした 3d プリンターの開発 [9]	17
20 第 3 章 仮説と提案	20
21 3.1 氷をマテリアルとした 3D プリンター	20
22 第 4 章 機構の実装	22
23 4.1 予備実験	22
24 4.2 造形の仕組み	24
25 4.3 液体窒素を用いた造形の実装	26
26 4.4 プリンターの本体	26
27 4.5 プリンターの制御	26
28 4.6 シリンジの機構	27

14.7	ノズルの機構	28
24.8	ベッドの機構	28
34.9	マテリアルの検討	29
第 5 章 開発したプリンターの精度調査実験		30
55.1	プリンターの動作検証	30
65.2	プリンター動作の確認	30
75.3	氷造形の初期実験	31
85.4	氷の積層実験	31
第 6 章 今後の展望		33
106.1	段落と改行	33
11	謝辞	34
12	参考文献	35

図 目 次

1

2	1.1 3D プリンターの造形方法一覧	2
3	2.1 3D ゲルプリンターで作製した造形物	5
4	2.2 実際に掘削して作製した高精度の彫刻の例	6
5	2.3 実際に印刷された布の造形物	7
6	2.4 ガラス造形の仕組み	8
7	2.5 造形されたガラスの造形物	8
8	2.6 制作中の様子	9
9	2.7 静電インクジェット式プリンターによるチョコレートプリントの仕組み	10
10	2.8 実際に吐出されたチョコレートを顕微鏡を用いて観察した様子	10
11	2.9 フルカラー 3D プリンターの概略図	11
12	2.10 2D プリンターとフルカラー 3D プリンターの違い	12
13	2.11 SLM 法を用いてセラミックスを印刷する際の様子の模式図	13
14	2.12 SLM 法を用いた印刷で完成した造形物	14
15	2.13 3D 食用ゲルジェットプリンタの模式図	15
16	2.14 寒天とゼラチンの溶液をそれぞれアルミの皿に垂らしたときの様子	16
17	2.15 積彩によって印刷した造形物	16
18	2.16 インクジェット型金属 3D プリンターの概略図	17
19	2.17 フロンガスを用いて作製した造形物	18
20	2.18 液体窒素を用いて印刷した造形物	19
21	4.1 設計図と造形した形	22
22	4.2 制作した実際の装置	23
23	4.3 造形過程と完成物	24
24	4.4 開発した氷をマテリアルとしたプリンターの全体図（前）	25
25	4.5 開発した氷をマテリアルとしたプリンターの全体図（後）	25
26	4.6 シリンジの機構	27
27	4.7 液体窒素造形用ベッド	28
28	4.8 砂糖を添加した割合ごとの水	29

1

第 1 章

2

はじめに

1.1 はじめに

3D プリンターは 2010 年代に低価格のものが登場するようになってから特に注目をされ続けて
いる。低価格化が進んだことで一般にも普及が進み 2020 年には 3D プリンター市場は世界で 210 億ドルになると予測されている。3D プリンターを活用することで製造業では、生産過程において開発期間やコストを削減することや素材の選択や高性能化ができる。特に、これまで大量生産により、均一化された商品が一般的な社会が形成されてきたが、3D プリンターの登場と普及により多種多様の者を少量生産ずつ、一般家庭で生産することができる。これまで、生産者と消費者は別の者であったが、3D プリンターの登場により生産者と消費者が同一の存在となるなりつつあるのだ。消費者の生産者化により、これまでにない発想の商品が数多く登場し、より便利なこれまでにない発想の商品はデジタル社会により、世界中に拡散され、人類社会の発展に貢献されると考える。3D プリンターは人類の可能性を最大化させるためのツールでもある。その 3D プリンターは印刷できる素材が限られているのが現状である。新たな 3D プリンターの素材を開発することは、多くの人が 3D プリンターを使い新しいものを作り出し、人類の想像力を最大化させるうえで重要なことだと考える。

1.2 現在の 3D プリンター

3D プリンターにはいくつかの種類がある。主に材料押出法、液槽光重合法、シート積層法、結合剤噴射法、材料噴射法、粉末床溶融結合法、指向性エネルギー堆積法などである。

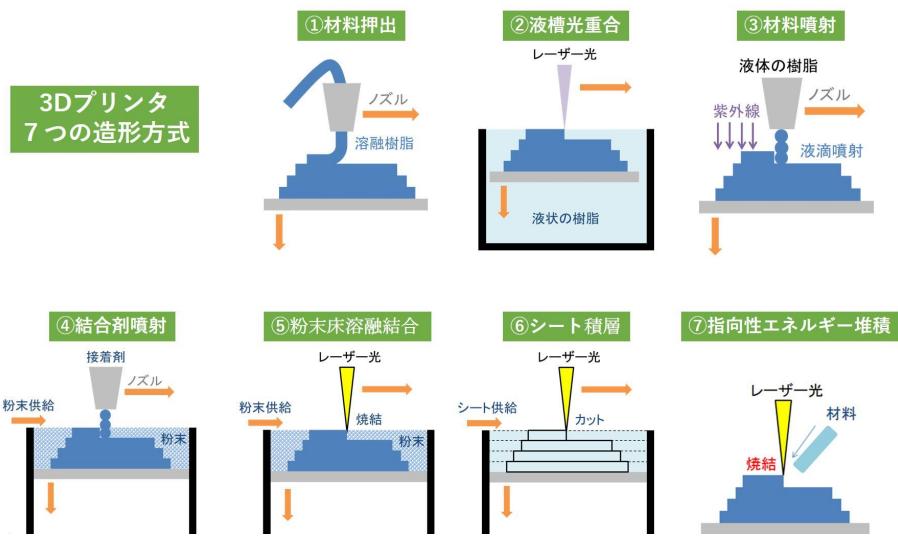
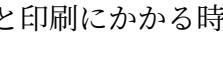


図 1.1 3D プリンターの造形方法一覧

<https://monoist.itmedia.co.jp/mn/articles/2007/28/news007.html>

- 1 特に一般に普及している 3D プリンターは 3 種類ある。
- 2 材料押出法中でも熱溶解積層方式 (FDM) は、ある程度の精度 ($\pm 0.1\text{--}0.2\text{ mm}$ 程度) と速度を有し、近年では 3 万円以下と低価格で一般にもっとも普及している。機構は、プラスチックのマテリアルを加熱し細いノズルから押し出して層を積み上げ造形物を作る。
- 3 液槽光重合法 (光造形法 (SLA)) は、最も古い 3D プリンターの積層方式で、紫外線で硬化する樹脂を使用し積層していく方法である。一層作るごとに樹脂が固まるまで紫外線を当てるため、造形に時間がかかる、その代わりに精度の高い造形 (0.05 mm程度) ができるのが特徴である。しかし、液槽光重合法 (光造形法 (SLA)) は、マテリアルの樹脂が高価であるとともに、造形後の掃除に手間がかかり、完成後も造形物が完全に硬化するまで紫外線を当て続ける必要がある。
- 4 粉末床溶融結合法、なかでも一ーザー焼結 (SLS) は、金属粉末にレーザーを当て熱で溶かすことにより積層を行う。レーザーで熱を加えるため時間がかかるが、高い精度でオブジェクトの印刷ができる。印刷には、金属を溶かすほどの出力の強いレーザーが必要になるためサイズやコスト、安全面を考えても一般家庭では使用が難しいのが現状である。しかし、鋳造では再現できない、液体を通すことのできる構造を作り出せるため、新しい活用方法も模索されている。

1 3D プリンターに重要とされる要素は印刷精度と印刷にかかる時間があると考える。それぞれの
2 プリンターがどの程度 2 つの要素を満たしているかを  に示す。縦の軸が精度を示し、横の軸が
3 印刷にかかる時間を表す。表を見てわかる通り、精度を優先すると速度が落ち、速度を優先する
4 と精度が落ちてしまう。このように、3D プリンターの印刷は時間と精度が反比例する。氷の造形
5 物を印刷するプリンターは、速度を犠牲にして正確な造形物を印刷する業務用タイプと造形を直
6 感的に行うためのハンディータイプのみで、中間の部分が欠落している。そこで、私は精度と速
7 度を両立させた氷の 3D プリンターを作成した。現在広く普及しているプリンターの一部を改修
8 することで氷造形を可能にする。

9 1.3 氷をマテリアルとした 3D プリンター

10 氷の彫刻は、世界中で様々なイベントやアート作品に用いられている。氷は透明なその美しさ
11 と時間とともに変化し、最後には溶けてなくなる儂さから人々に親しまれており、様々な作品が
12 作られている。しかし、誰でも簡単に触れ合えるものではなく、氷の作品を楽しめる場所は限ら
13 れているのが現状である。その原因として氷の作品は作るのに時間がかかり、彫刻の技術や設備
14 が必要となる。私は誰でも簡単に思い通りの形状の氷の作品を作れるようするために、3D プリン
15 ターを使い氷の造形物をプリントする新しい手法を提案する。現在の 3D プリンターによる高精
16 度な氷造形は、20mm/h のスピードで高さ 0.1mm の積層をしていく。この速度で氷の造形物を
17 作るには 0 度以下の部屋を用意し、造形中は常に造形物の周りを低温に保っておかなければなら
18 ない。また、Suntory-3D on the Rocks では CNC を使った切削のため特殊な機材と知識が必要
19 になるため、だれもが扱えるものではない。3D プリンターとして開発されるだけでなく、一般
20 の多くの人に普及させるためには精度を保ちつつ素早く造形できるプリンターである必要がある。
21 私は、3D プリンターを使う知識がある人ならば、スキルに依存せずに氷の造形物を作ることがで
22 き、現状の氷をマテリアルとした 3D プリンターよりも高速で造形のできる 3D プリンターを提
23 案する。

1.4 氷造形の提案

氷の造形物をつくる試みは過去に様々の方法で試されてきた。氷をマテリアルとした自動造形には現状 2 パターンがある。1 つ目が氷の塊を CNC で掘削する方式である。CNC 方式では、精度の高い氷をマテリアルとした造形を行おうことができるのだが、特殊な設備と技術が必要になる。2 つ目が 3D プリンターと同じ仕組みで精度の高い造形物を印刷することができるが、造形速度が 20mm/h のスピードで高さ 0.1mm の積層とかなり速度がかなり遅い。これらの氷をマテリアルとした造形方法では造形速度が遅く、冷凍庫のような特殊な環境が必要になるため、一般の人気が気軽に氷をマテリアルとした造形を利用することが難しいのが現状である。氷をマテリアルとした造形物の使用用途は、現実に氷をマテリアルとした工業製品が無いのを考えると、強度的な問題や常温で溶け出す問題などにより、観賞用に用いられることがほとんどと考えられる。使用用途が観賞用であると考えた場合、工業用製品に必要な要素が正確性（精度）などに対して、観賞用では、氷の条件を満たしたうえで一般の人でも扱いが可能かつ、一般のユーザーが設計したデータにできるだけ近い形に印刷されれば問題ないと考えられる。また、氷を再定義する。氷をまたリアルとした 3D プリンターの主な使用用途は観賞と考える。観賞とは、氷特有の常温で液体へと徐々に変化し、最後にはなくなるこの過程が、ほかのマテリアルにはない大きな特徴であるこの部分を指すと考える。そのため、氷の定義として下記の 2 つを定義する。

1. 固体であるとき常温より低温であること。

2. 常温で溶けること。

氷の定義を以上のようにしたうえで、一般の人でも扱いが可能かつ、一般のユーザーが設計したデータにできるだけ近い形に印刷できる氷をマテリアルとした 3D プリンターの提案する。

1

第 2 章

2

関連研究

3 2.1 関連研究

4 新しいマテリアルを使い、今までにない 3D プリンターでは表現できなかったモノを作ること
5 を可能にしている研究を中心最新の 3D プリンターに関する研究を調査した。

6 2.2 ゲルを用いて印刷する 3D プリンター [1]

7 この研究は、ゲルをマテリアルとして用いた造形物を 3D プリンティングするものである。図 1
8 のように、ゲル溶液を使用しながら強度や感触を部位によって変化させて造形物を作成すること
9 ができる。

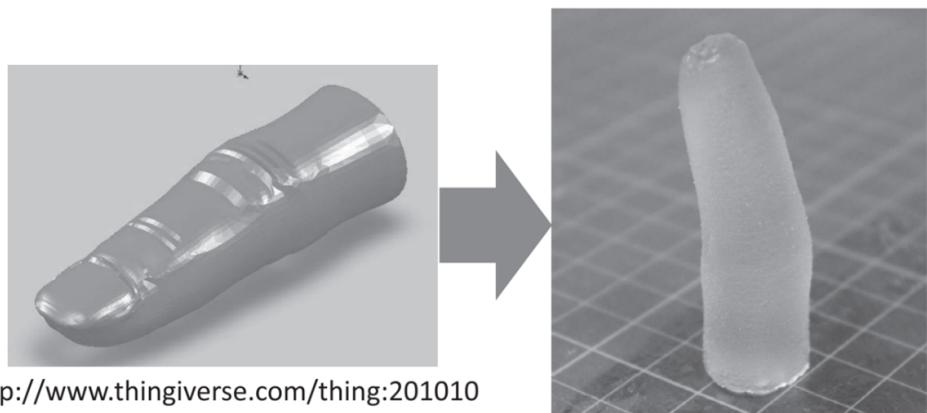


図 2.1 3D ゲルプリンターで作製した造形物

10 これは、ゲル化を誘起する UV レーザーを光ファイバーを通して局所的にゲル溶液に照射する
11 ことで、ゲルの 3 次元造形を可能にしている。3D プリンターは現在、臓器の立体イメージを作り

1 出すのに医療分野で活用されているが、手術の計画や事前検証のための立体の臓器モデルを作製
2 するには、数千万もする工学な 3D プリンターを使用してプラスチックやゴムなどの、実際の臓
3 器よりもはるかに硬い樹脂を用いて造形をする方法しか存在しなかった。このゲルを用いて印刷
4 する 3D プリンターは、低コストで感触がより患者のものと似ている臓器モデルを作成できる可
5 能性を秘めている。

6 この 3D プリンターは材料として微粒子調整ダブルネットワークゲル(略称:P-DN ゲル)を使用
7 している。このゲルは強電解質性を示すモノマー由来の堅く脆い高分子ネットワーク(1st ネット
8 ワーク)と、中性を示すモノマー由来の柔軟な高分子ネットワーク(2nd ネットワーク)が相互侵
9 入網目構造をとっている複合材料である。この P-DN ゲル溶液に UV レーザーを照射することで
10 ラジカル反応が生じ、ゲルの 3 次元構造をつくることができる。

11 2.3 Suntory-3D on the Rocks[2]

12 氷を掘削し様々な彫刻を作りお酒に入れて楽しむ試みがある。多軸の CNC を使い掘削するこ
13 とで高精度の彫刻を作ることができるが、一般に普及させるのはコストならびに加工中の冷却の
14 面から考えると難しい。

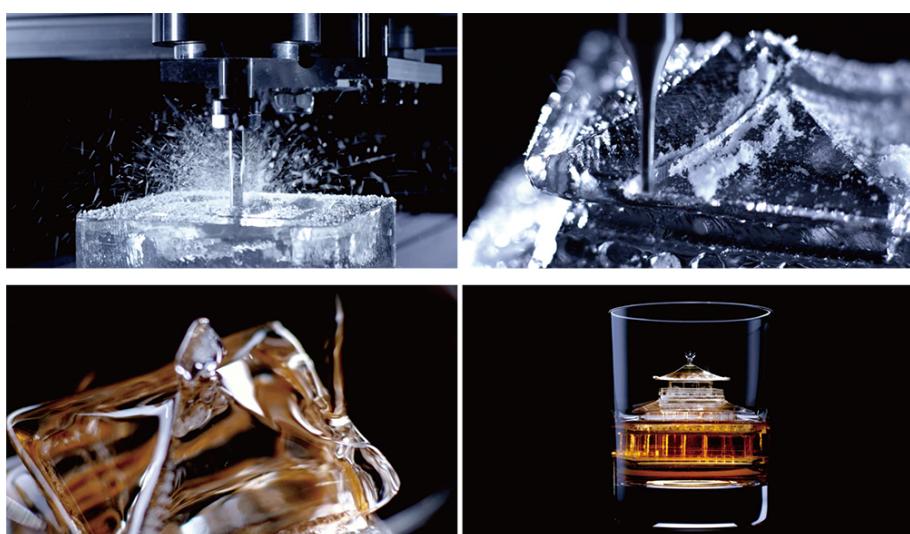


図 2.2 実際に掘削して作製した高精度の彫刻の例

<https://mag.sendenkaigi.com/brain/201406/up-to-works/002420.php>

2.4 A Layered Fabric 3D Printer for Soft Inter-active Objects[3]

この研究では布の造形物を印刷するためのプリンターを紹介している。また、布の中に導電繊維の層を入れたり、コイル状の布を入れることで、タッチセンサーとしての利用法や NFC を使い LED を光らせるアプリケーションが紹介されている。このプリンターの仕組みは、布のロールを引き出し天板に吸着させ固定し、レーザーでモデルの輪郭を切断し布のロールから切り出す。この工程を繰り返し、アイロンの熱で接着していくことで造形物が完成する。

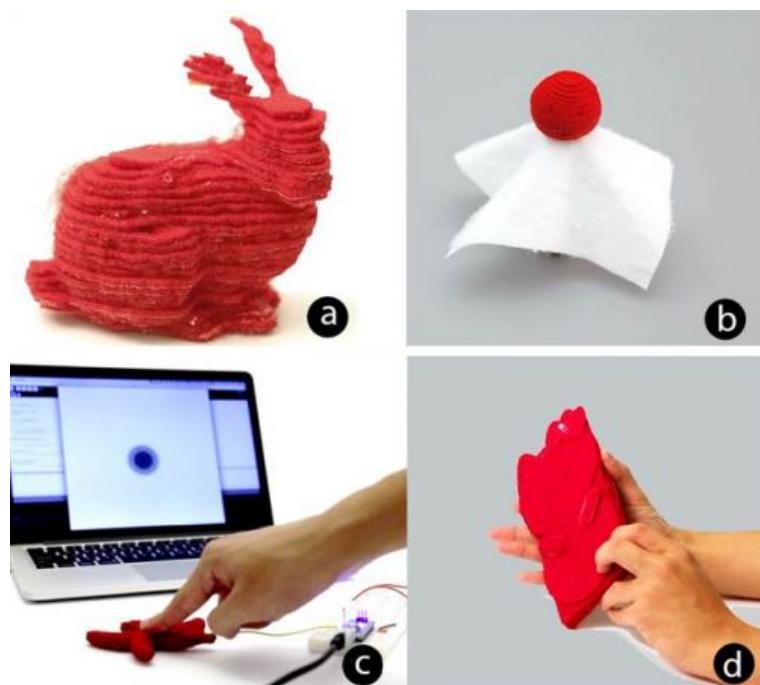


図 2.3 実際に印刷された布の造形物

[https://thelastnewspaper.com/
a-layered-fabric-3d-printer-for-soft-interactive-objects/](https://thelastnewspaper.com/a-layered-fabric-3d-printer-for-soft-interactive-objects/)

2.5 Additive manufacturing of optically trans-parent glass[4]

ガラスをマテリアルに使ったこの研究では、高温で流動性の高い状態に保持されたガラスを貯めておき、そのガラスを垂らすことで造形していく。実際に造形されたガラスの造形物は一回の

- 1 ストロークで出せるラインは太く分厚いものであり、細かい造形はできないがサイズの大きい花瓶のようなものを造形することができる。この手法で造形された花瓶は光を乱反射させる特性があり、上からライトを当て光の波紋を楽しむアプリケーションが提示されていた。

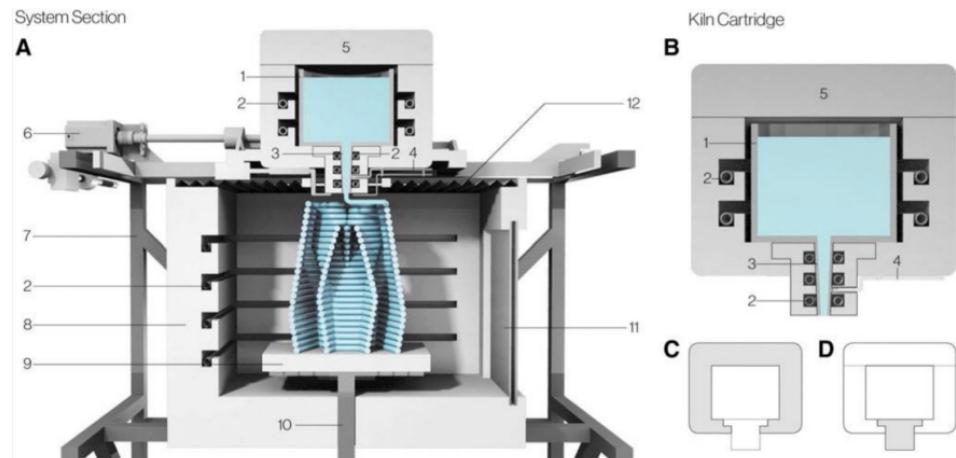


図 2.4 ガラス造形の仕組み



図 2.5 造形されたガラスの造形物

<https://www.behance.net/gallery/65276297/GLASS-I>



図 2.6 制作中の様子

[https://www.solidsmack.com/fabrication/
mits-mediated-matter-group-unveils-transparent-glass-3d-printer/](https://www.solidsmack.com/fabrication/mits-mediated-matter-group-unveils-transparent-glass-3d-printer/)

2.6 静電インクジェット式 3D プリンタによる高粘度食品材料の高精度プリント [5]

この研究では静電インクジェット法を用いることで、高い印刷精度で高粘度材料を用いた、視覚的、味覚的に優れた食品の印刷を可能にする 3D プリンターの開発をしている。

従来の食品の 3D プリンターには熱溶解式 (FDM) 式 3D プリントを用いたチョコレートの印刷がある。しかし、熱溶解式では積層ピッチが約 0.5[mm] 以上と非常に粗く、また高粘度材料をプリントする際には添加物を加える必要がある。この添加物には、食品の味に影響が出てくる問題点がある。静電インクジェット法を用いると、この問題を解決すると同時に高精度な印刷が可能となる。下の模式図の様なチョコレートプリンターを作製し、実験を行った。

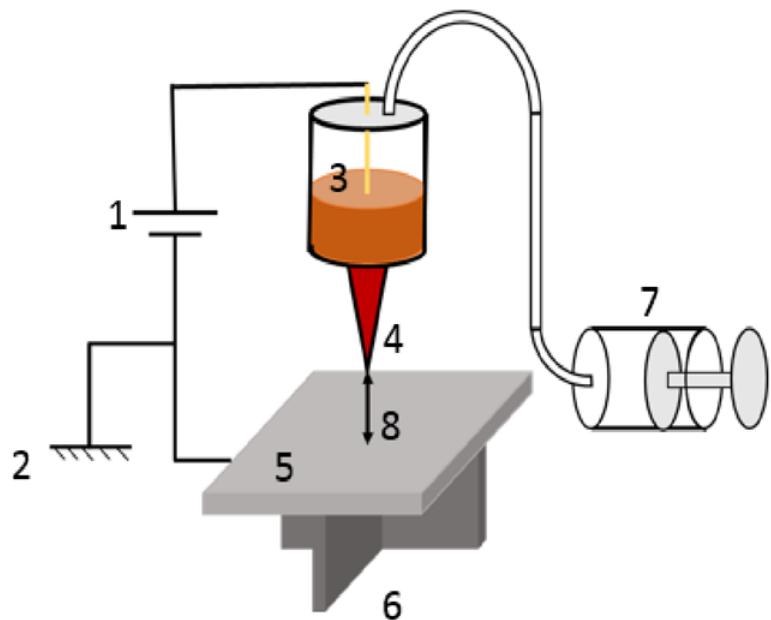


図 2.7 静電インクジェット式プリンターによるチョコレートプリントの仕組み

- ¹ この 3D プリンターは高粘度食品材料であるミルクチョコレートに高電圧を加え微小液滴を吐
- ² 出し、下図のような超微細なラインを印刷することを可能にする。また電圧のコントロールに
- ³ よって、吐出するラインの径を制御することもできる。

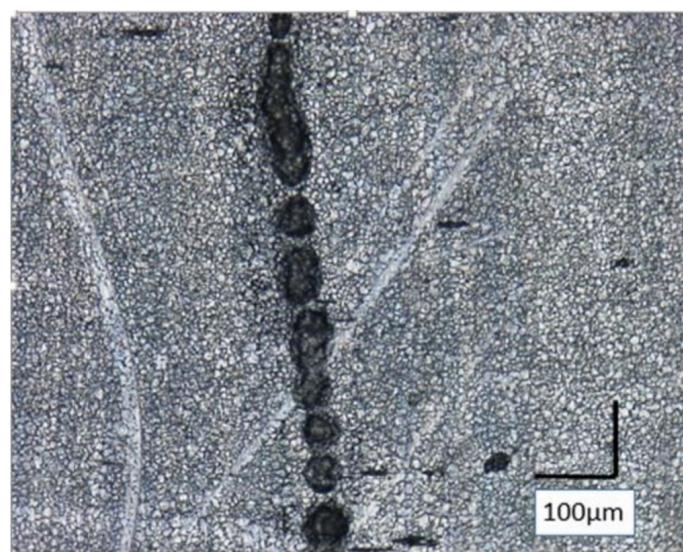


図 2.8 実際に吐出されたチョコレートを顕微鏡を用いて観察した様子

2.7 フルカラー 3D プリンター—2D 印刷から 3D 印刷へ—[6]

フルカラー 3D プリンターについて紹介する。このプリンターは UV(紫外線) 硬化インクジェット方式を採用したことで任意の 3D 形状の造形を可能にすると同時に、その表面にフルカラーで印刷することができる。このフルカラー 3D プリンターにより、クリエイターの創造物が画面の中だけではなく、今までより容易に手に取れるようになっている。新しい市場も徐々に出現し始めていて、最近では 3D 撮影による人文やペットのリアルなコピー造形物が話題になっている。

この 3D プリンターは、下図のようにインクを UV 光源で硬化させながら積層法で造形を行う。

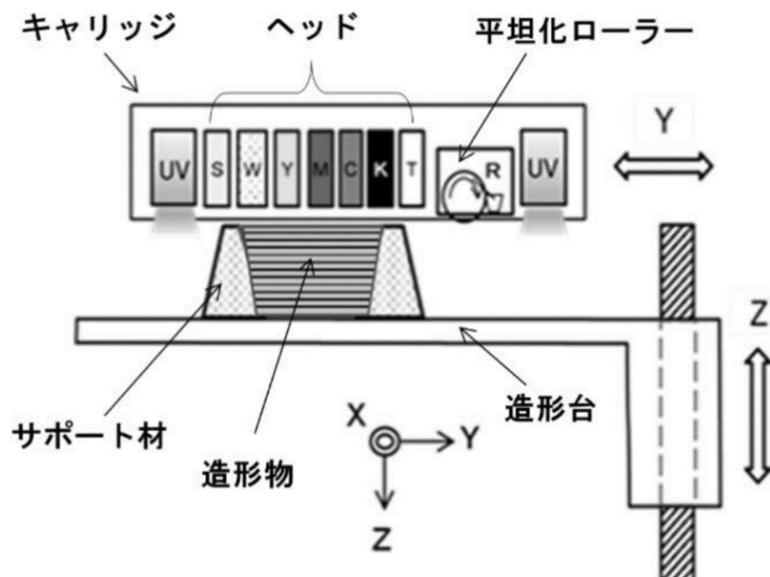


図 2.9 フルカラー 3D プリンターの概略図

2D と 3D を比較した時、フルカラー 3D プリンターでは下図のような印刷が行われている。

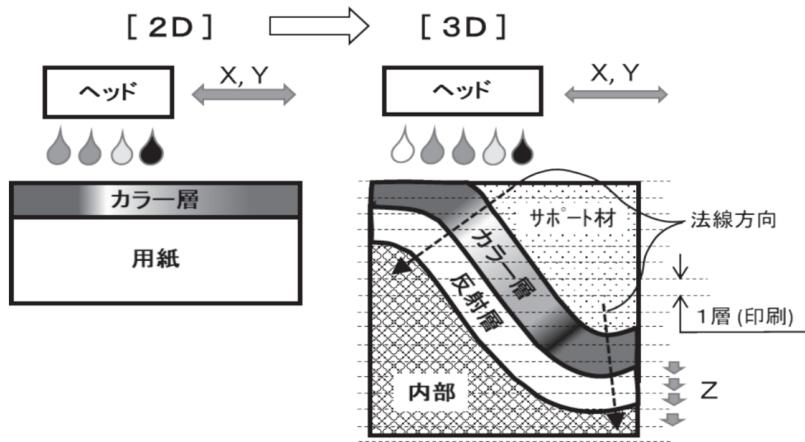


図 2.10 2D プリンターとフルカラー 3D プリンターの違い

- 1 2D の印刷では画像データの濃度によりカラーインクの量が変化するが、これを 3D に適用する
- 2 とカラー創の外形が崩れてしまうため、カラーインクのない空きスペースには透明インクを補填
- 3 して外形を保つ工夫をしている。

4 2.8 3D プリンタのセラミックスへの適用 [7]

- 5 この研究では、セラミックス材料を用いた 3D プリンター開発を行っている。主にセラミック
- 6 施密体を作製する基礎検討に関する研究である。
- 7 今回このプリンタには、SLM 方式を採用する。この方式では下図のように、薄く敷き詰めた粉
- 8 末床にレーザや電子ビームを走査して粉末を溶解し、順次積層することで 3 次元の造形物を得る。
- 9 昨今では、レーザや電子ビームの出力向上に伴い、新規な材料の適用が可能となってきたが、
- 10 セラミックスに関しては、急熱急冷を伴うプロセスの特性上衝撃熱が発生するため構造体の密度
- 11 向上が難しく、工業的な部材の製造は実現していない。この高密度焼結体の迅速な 3 次元造形が
- 12 実現すれば、小ロットの射出成形やテープ整形の代替、複雑形状を生かした高性能セラミックス
- 13 フィルターや半導体作製用の露光ステージへの利用が期待できる。

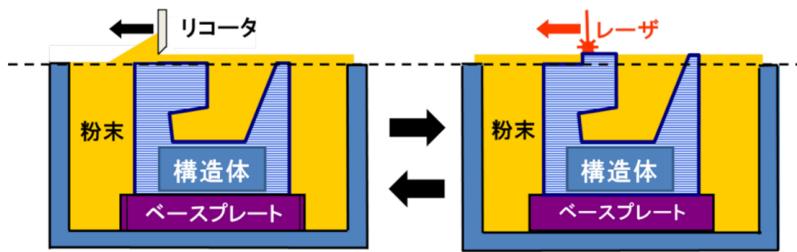


図 2.11 SLM 法を用いてセラミックスを印刷する際の様子の模式図

- 1 熱衝撃を回避したセラミックスの SLM 法として、直接セラミックスを焼結せずに、レーザによる
 2 形状の作製と焼結による密度向上を分離した間接法が考案されている。この間接法では、セ
 3 ラミックスと低融点の樹脂成分と複合化した粉末を用い、樹脂部分のみをレーザ溶解することで
 4 シート成形、グリーン体を作製し、その後、脱脂・焼結することによりセラミックス単体の焼結
 5 体を得るものである。間接法を用いた様々な試みがなされてきたが、造形に時間がかかる、密度
 6 が低いなどの理由で工業的な利用には未だ至っていない。
- 7 この研究では、高強度アルミナ焼結体の作製を目的とした間接法プロセス構築のための検討を行
 8 い、アルミナの相対密度が 94 % の焼結体の作製に成功している。検討のため、①～③のそれぞれ
 9 の特性に着目した
- 10 ①原料セラミックスの選定・粒子径・粒子径分布
 11 ②原料樹脂の選定・脱脂性・樹脂の融点またはガラス移転点
 12 ③造粒粉の作製・流動性・粒子径・かさ密度・1 個粒子の密度
- 13 その他にも、SLM 条件の最適化、3D プリンタ内の粉体挙動のシミュレーションを行った結果、
 14 以下の図のような、アルミナの相対密度が 94 % セラミック緻密体を作製することに成功した。

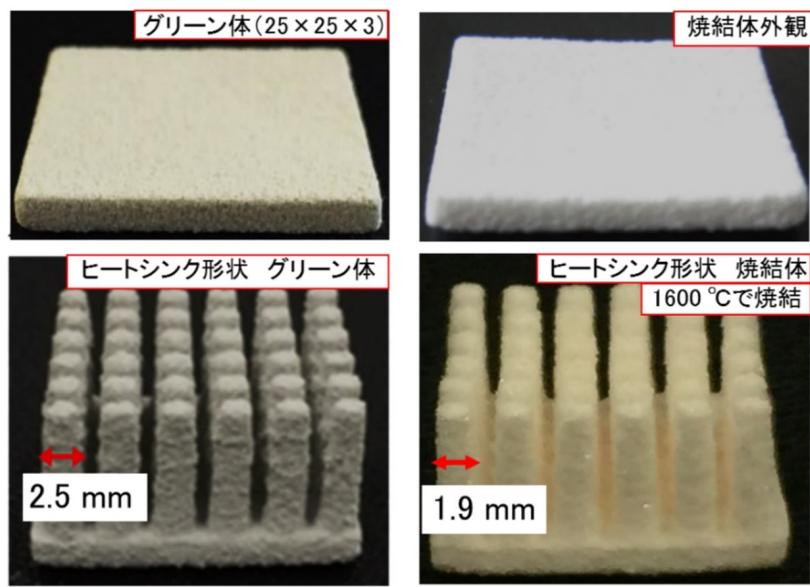


図 2.12 SLM 法を用いた印刷で完成した造形物

2.9 3D 食用ゲルジェットプリンタによる食品創製 [8]

この研究では、現代の高齢者が食事をより食べやすく、見た目も楽しめるような食品が造形できる、食用のゲルプリンタの開発をしている。

この研究で使用するゲルプリンタは下の概略図のように、①搬送系、②液送系、③冷却系の、三つの機構から成り立っている。

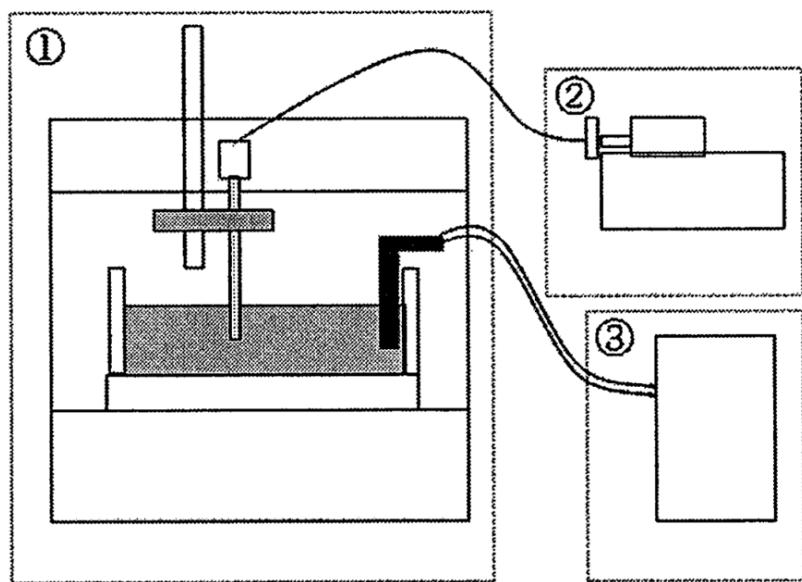


図 2.13 3D 食用ゲルジェットプリンタの模式図

①は、②のシリンジポンプによって送り出された溶液をコンピュータ制御により、任意の形状に造形することが可能な機構である。②は、ゲルの溶液をシリンジで①の方へ押し出す機構となっている。③は、溶液を滴下する水槽を冷却する機構である。この装置を用いることで、アルギニン酸ゲル(人工イクラ)の造形が可能となる。

またこの研究では、プリンタに使用するゲルの冷却方法の検討も行った。寒天とゼラチンの溶液をシリンジを使用しアルミの皿に垂らしていった。1分経過後、寒天、ゼラチン共に固まったが、寒天に比べゼラチンは少し皿にくっついて剥がそうとするとボロボロになった。二つに共通し濃度が高くなると固まりやすくなる傾向があるが、その場合でもゼラチンは強く付着する傾向がある。これは、寒天とゼラチンが植物性と動物性という違いに起因するものだと考えられる。

下図が寒天とゼラチンが固まった様子である。

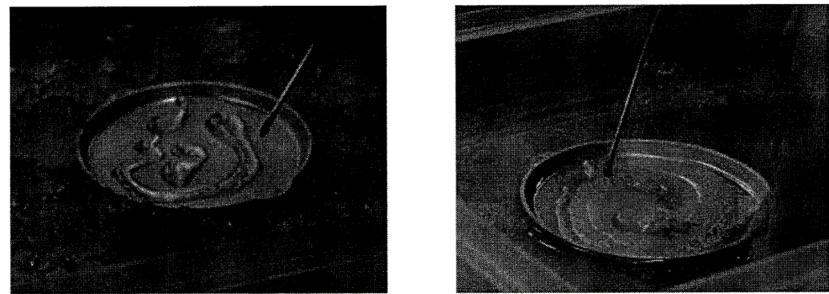


図 2.14 寒天とゼラチンの溶液をそれぞれアルミの皿に垂らしたときの様子

1 2.10 積彩 [9]

2 通常の 3D プリンター特にカラー 3D プリンターでも複数の色をせきそうすることでカラー 3D
3 プリンターとしている。この積彩では、コンピューティングによって調色しながら色糸を積む 3D
4 プリンティングの製造によって造形・着彩をひとつの工程として扱っている。これを「積彩」と
5 呼び、また、繊細な織物のように糸を積んでいく積彩は新たな表現技法（虹のように変化する色
6 彩効果）を可能にし、私たちはこの技法を応用して「色瓶」というプロダクト製作している。



図 2.15 積彩によって印刷した造形物

<https://idarts.co.jp/3dp/toyamadesign-gp-color-fab/>

7 2.11 放電現象を利用したインクジェット型金属 3D プリンター開 8 発に関する基礎研究 [10]

9 この研究では、熱可塑性樹脂を用いた材料押出型の 3D プリンターの機構をベースとした金属
10 3D プリンターの開発をしている。従来の金属材料を扱うことのできる 3D プリンターの例とし

て、積層造形、粉末床溶接接合、結合剤噴射、溶接肉盛などを利用したものがあるが、これらの方法は装置も大型で価格も非常に高い。また、金属粉末の結合力が十分ではない指摘もされている。材料押出型で金属を扱える3Dプリンターが開発されれば、低価格な金属3Dプリンターが実現できる。

下に3Dプリンターの概略図を示す。この3Dプリンターには①細線繰り出し電極、②薄肉パイプ回転電極の二つの特徴がある。

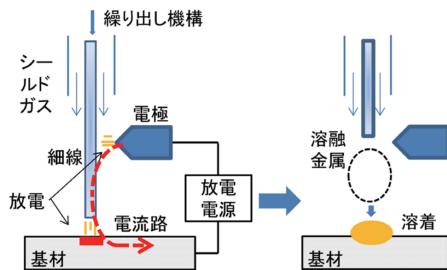


図2.16 インクジェット型金属3Dプリンターの概略図

2.12 Elsa:氷を素材とした3dプリンターの開発 [9]

この研究は造形速度と造形精度を両立させ、一般に普及している3Dプリンターと同じ学習コストで使える氷をマテリアルとした3Dプリンターの開発を行っている。1つ目の手法として液化した代替フロン(HFC134a)を使用する方法を提案している。フロンが断熱膨張する際に周囲の熱を奪うのを利用して水を冷やし、瞬時に氷を作る。エアーブラシを使い水とフロンガスを噴射し氷を作る機構を実装した。しかし、3つの問題点がある。1つ目は、コストが高いことである。3つ目は、造形物がフロンガスを含み純粋な氷ではないこと。3つ目は、環境に対して悪影響があることである。



図 2.17 フロンガスを用いて作製した造形物

- 1 2つ目の手法として液化した代替フロン (HFC134a) を使用する方法の問題を解決できる氷を
- 2 マテリアルとした 3D プリンターとして、液体窒素を使用する手法を提案している。



図 2.18 液体窒素を用いて印刷した造形物

第 3 章

仮説と提案

2

3.1 氷をマテリアルとした 3D プリンター

4 3D プリンターに実装した、氷を作るための機構について述べる。これまでの氷の造形方法は大
5 きく分けて 2 つある。大きな氷から切削して造形するもの。もう一つが、氷を少しづつたらし長
6 時間をかけて、造形するものがある。どちらも取り扱いが難しく、造形するのに長時間をしてし
7 しまうのが問題だ。また、短時間できる氷の造形として、過冷却水を使っての造形が有名である。
8 しかし、過冷却水の場合準備に時間がかかる上、温度変化に敏感で少しの衝撃でも凍り始めてし
9 まうため、制御が難しい。一般の人でも扱いが可能かつ、一般的なユーザーが設計したデータにて
10 きるだけ近い形に印刷できる氷をマテリアルとした 3D プリンターの提案する。必要な要素とし
11 ては以下のようである。

- 12 1. ある程度の精度で造形ができること。
- 13 2. 通常の 3D プリンターと同程度の速度で印刷ができること。
- 14 3. 氷の定義を満たしていること。
- 15 4. 3D プリンターが扱える人であれば、短時間で扱えるようになること。

16 それぞれの要素について実装にするにあたり、以上のことが有効ではないかと考える。「ある程
17 度の精度で造形ができること。」「通常の 3D プリンターと同程度の速度で印刷ができるこ。」を
18 満たすために液体窒素を使った造形方法が有効ではないかと考える。また、他の研究では、特殊
19 な機材を使用し、装置が高価になりがちである。液体窒素は、日本各地で手に入る上、価格も 1

1 リットルあたり 300 円と安価であるため、今回の研究で使用することにした。「3D プリンターが
2 扱える人であれば、短時間で扱えるようになること。」を満たすためには、既存の 3D プリンター
3 と同様の使用方法で使える必要があるため、世界中で使用されている 3D プリントおよびスライ
4 サーソフトウェアである Ultimaker Cura で操作が可能である必要があると考える。氷の定義を
5 したことで、純粋な水以外でも氷の造形ができる。純粋な水を積層する場合、水の粘度が低いた
6 め、固まる前に広がってしまう。そのため造形精度が悪く、造形物のからはみ出した部分には造
7 形ができず、オーバーハングなども造形することが難しい。よって、水の粘度を上げることによ
8 り、上記の問題を解決できるのではと考える。また、粘度を上げる手段として、いくつかの方法
9 が考えられる。水に砂糖などを加え粘度を上げる方法とシャーベット状のものをマテリアルとし
10 て使用する方法だ。シャーベット状のものを使用する場合は、温度管理が必要になるため、今回
11 の機構では、砂糖を加え粘度を高めたものをマテリアルとして使用する。

1

第 4 章

2

機構の実装

4.1 予備実験

水に砂糖等の粘度を上げられる物質を添加し造形を行う方式の実証を行った。初めに、3D プリントとして自動化させる前に、水の粘度が造形物の造形速度、造形精度、オーバーハングの造形に影響を与えるのか調査を行った。造形の仕組みは図のようになっている。実験の装置は、保温のため一番下に発泡スチロールの容器を用意した。その上に-196 度の液体窒素を十分に注ぎ、さらにその上からアルミトレーを沈める。それにより、アルミトレーも液体窒素に近い温度まで冷やされ、そこに注射器を使い水あめと水の中間の粘度の水をたらすことで、冷やされた水が氷に変わる。冷やされた氷の温度は 0 度よりも低く、その上に水をたらすと氷柱ができるように氷が積層される。

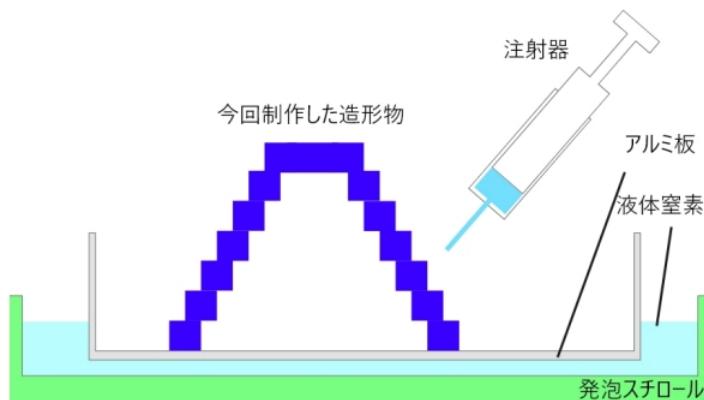


図 4.1 設計図と造形した形



図 4.2 制作した実際の装置

1 制作した装置は図 4.1 である。この装置を使い、水の粘度がどの程度、造形速度と造形精度に影
2 韻するのか調査を行った。造形物はオーバーハングの調査を行うため図のように中を空洞になる
3 ように造形を行った。実際に制作した装置を使い造形を行ったものが図である。造形時間は約 5
4 分ほどで完成した。造形精度の問題もあるが通常の 3D プリンターよりもかなり早い結果になっ
5 た。大きさは横幅約 3 センチ、高さ約 1.5 センチほどである。使用した水の量は、約 200ml ほど
6 である。初めに想定した形通りに造形ができ、オーバーハングの造形も成功した。また、発見し
7 た特徴として、透明度の高い氷を制作することができた。



図 4.3 造形過程と完成物

結果は予想通り、造形速度の改善と、オーバーハングができない問題の解消、これら二つを改善しつつ、さらに造形精度の向上ができた。通常のプラスチックをプリントする 3D プリンターと比較して、押し出される水の粘度が関係していることが分かった。また、造形する際に使用した水が砂糖を溶かすために加熱していた。この余熱があったため、注射器から押し出すときの温度が 50 度くらいになっていた。そのため、液体窒素の注射器内部の水が冷え固まらず、すでに造形されている造形部分の表面を溶かすため、造形物が水をはじくことなく接着できているのではないかという仮説が仮説を立てることができた。

4.2 造形の仕組み

予備実験では、液体窒素と水に砂糖を混ぜ粘度を上げることにより、ある程度の精度と速度を持つ事が分かった。ここでは、予備実験を自動化させ、プリンターが氷を積層造形していく仕組みについて解説する。造形用のペットに熱伝導率の高い金属製のアルミプレートを使用し、液体窒素の保温性を高める為発泡スチロールでできた容器に沈めた。液体窒素は-196 °Cであり、アルミプレートもそれに近い温度まで冷やされる。そこに水をたらすことで、水が冷やされ氷が作られる。氷は、アルミプレートを通して、液体窒素により冷やされ続けるため、氷の温度も 0 °C以下になる、その上に水をたらすとその水も氷へと状態が変化し氷が積層される。

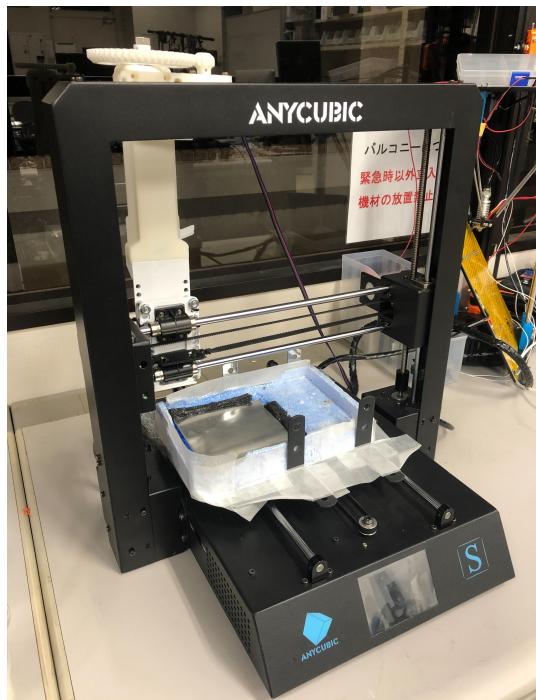


図 4.4 開発した氷をマテリアルとしたプリンターの全体図（前）

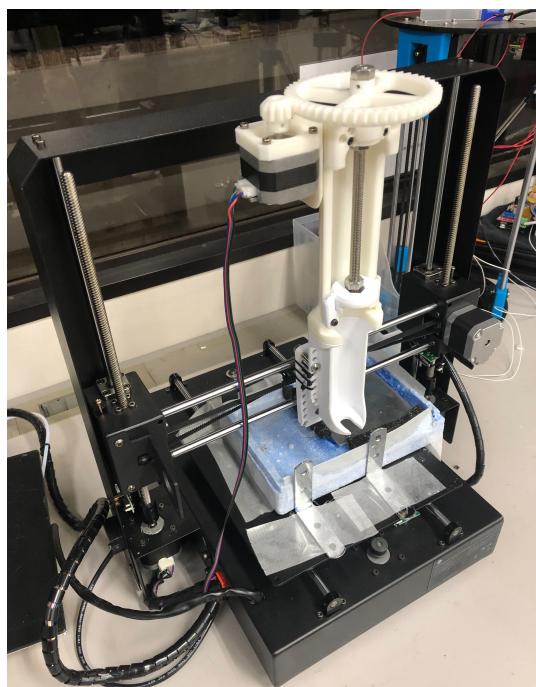


図 4.5 開発した氷をマテリアルとしたプリンターの全体図（後）

4.3 液体窒素を用いた造形の実装

ここでは、液体窒素を使用した造形機構について述べる。機構の全体像は図のようになってい
る。ノズルから水を供給するためのシリンジを押し出す機構を実装した。また、今回の水は粘度
を持たせているため、長いチューブを用いてしまうと、抵抗で押し出すのが難しくなる。そのため
シリンジからノズルまでの距離ができるだけ短くなる機構を実装した。

4.4 プリンターの本体

3D プリンターも一般に販売されているものを改造して使用した。使用した 3D プリンターは、
「Anycubic i3 Mega S」である一般に売れれている 3D プリンターの改造で氷をマテリアルとし
た造形ができれば、3D プリンターが使える一般の人が短期間に学習が可能だと考える為である。
また、大半が既存の部品であるため、一般への普及もしやすいと考えるためである。

4.5 プリンターの制御

氷の 3D プリンターの制御は、Marlin-Ai3M という 3D プリンターの制御用アプリケーション
と Marlin というファームウェアを一部改造し使用している。改造内容は、モーターの駆動方向
の変更、モータードライバーへの対応、3D プリンターは安全装置として、一定の温度以下で作
動しないようになっている。この安全装置が 0 °C 以下で稼働する氷の 3D プリンターでは必要が
無いため、無効にさせた。今回制作したプログラムは、Ultimaker Cura を通して「Anycubic i3
Mega S」にアップロードさせた。この作業を行ったことにより、基本的に一般に販売されている
3D プリンターと同じように制御することができる。

4.6 シリンジの機構

シリンジを押し出して水を供給するために、既存のエクストルーダー用のモーターを利用して
いる。シリンジを押し出すためにモーターの回転を上下の運動へ置き換えるために、全ねじ棒を
使用した。また、そのままモーターを直結してしまうと、力不足になることが想定されてため、ギ
ヤで回転数を調整している。既存の3Dプリンターでは、フィラメントを押し出すモーターを利
用しているため、PCを使い水の押し出し量を自由に調整することが可能であり、安定して造形が
できる設定を模索することができる。また、シリンジは3Dプリンターを使い制作したが、サイズ
が研究室にあるプリンターに収まりきらなかったため、上下に分割して印刷し、印刷後接着剤に
より合体させている。

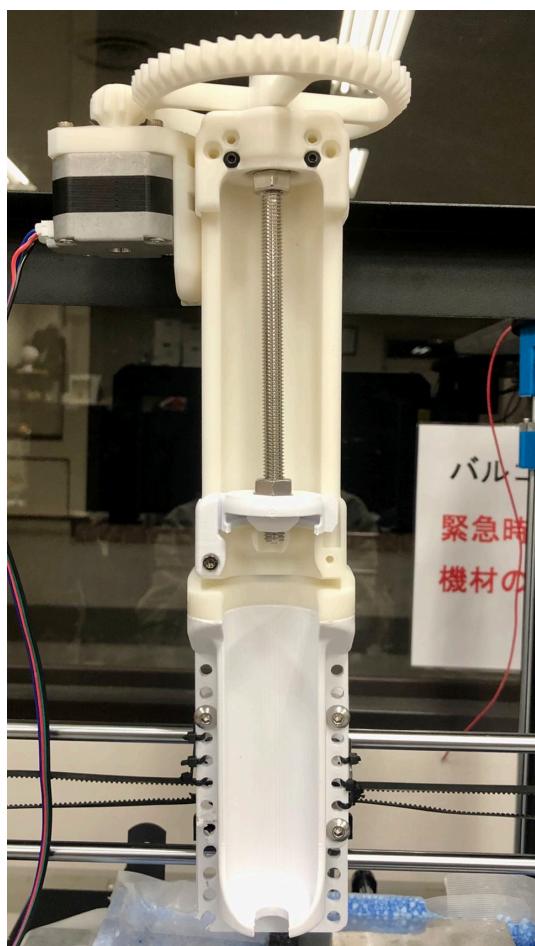


図4.6 シリンジの機構

4.7 ノズルの機構

水に砂糖を加え、粘性を持たせた液体を流すため、シリンジからノズルまでの距離が長いとその間で抵抗が発生しシリンダーやシリンダー制御用のモーターに負担をかける。そのため、できるだけ、シリンジからノズルの距離が近い方がいいと考える。そこで、シリンジを直接ノズルとして使用し、素材を押し出せる機構を制作した。また、ノズルサイズは今回の実験では 1mm の穴を空け試行する。

4.8 ベッドの機構

液体窒素で氷を造形するベッドとして、大きく 2 つに分けられる。1 つ目が液体窒素用のトレーだ。液体窒素用のトレーでは、下に保温性を高め、液体窒素の持ち時間を長くするために、発泡スチロールを使用した。また、アルミプレートの下に液体窒素がたまるようにくぼみをつけている。2 つ目が造形用のトレーだ。熱伝導率の高い金属のプレートを使用する。今回はアルミ製のプレートを使用した。また、造形したものを取り出しやすくするために、取り外しが容易な設計を行った。完成したハードは図 4.8 のようになっている。



図 4.7 液体窒素造形用ベッド

4.9 マテリアルの検討

今回開発する 3D プリンター FDM 方式の改良である。FDM 方式では、常温で固体の物質を加熱し柔らかくしてから押し出している。水のプリントの場合、既に液体を押し出し、冷やして固体にしてプリントする。違いとして、押し出されるときのマテリアルの粘度が印刷精度や印刷時間に違いをもたらしているのではと考える。今回開発した氷をマテリアルとして 3D プリンターの材料として水に砂糖を混ぜて粘度を上げたものを用意した。砂糖はどこでも手に入りかつ安価で扱いも簡単であるため、今回の実験に採用した、水と砂糖の割合は 1:1, 1:2, 1:3 のもの用意した。また、1:2, 1:3 の液体を作るにあたり、鍋で一度加熱する必要がある。

水：砂糖

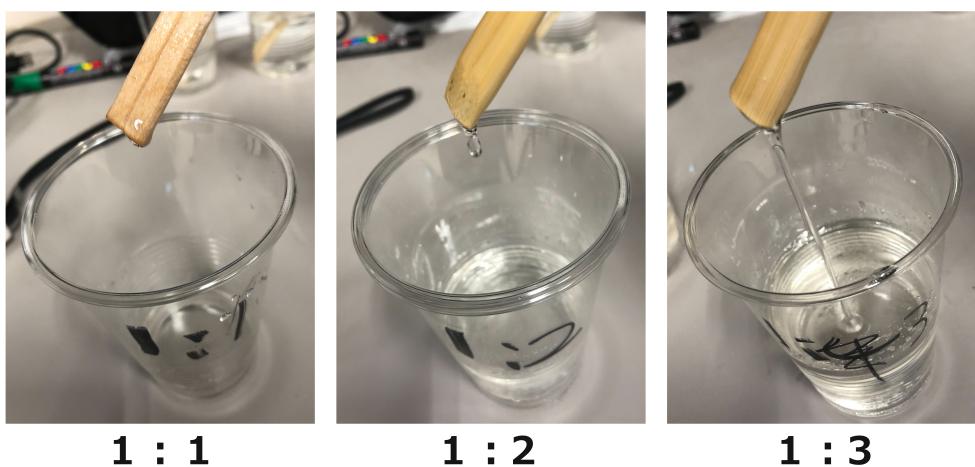


図 4.8 砂糖を添加した割合ごとの水

1

第 5 章

開発したプリンターの精度調査実験

5.1 プリンターの動作検証

4 始めに、開発した氷をマテリアルとした 3D プリンターの動作テストを 2 つに分けて行った。1
5 つ目が、プリンターの動作テストだ。2 つ目が、氷を積層できるかの調査を行った。この二つの検
6 証をこなうことにより、プリンターの挙動や造形の特徴を知ることができる。

5.2 プリンター動作の確認

8 開発した氷をマテリアルとした 3D プリンターの動作テストを行った。一般に販売されている
9 FDM 方式の 3D プリンター「 Anycubic i3 Mega S 」を改造して使用している。ノズルを通常
10 の取り付け位置に設置してしまうと、ノズルの上部が 3D プリンターの上部と接触してしまうた
11 め、今回はノズルを左右を繋ぐポールの後ろ側に設置し、そのうえで、ポールから 1.5 cm 離してい
12 る。そのため、Ultimaker Cura で表示されている、印刷が可能な位置と比べ 5cm 後ろにずれて
13 いる。また、開発した氷をマテリアルとした 3D プリンターの設計上アルミプレートの上でしか
14 造形ができない。そのため、実際に造形できる範囲は 100mm × 80mm × 250mm であることが
15 分かった。

16 シリンダーの押し出し機構については、基本的には問題なく作動した。しかし、3D プリンター
17 が作動していないタイミングでも、ノズルの先から水が漏れ出してしまったのが問題として浮上し
18 た。また、おそらく水が漏れ出したことに起因して、シリンダー内部の圧力が低下した。この影
19 響により印刷を始めた際にしばらくノズルから水が出ない問題が発生することが分かった。しか

- ¹ し、現状 Ultimaker Cura の設定により、始めにシリンダーを少し押し出す設定になっている。
- ² これにより印刷には影響していないが、対処的な解決策であり、根本的な解決が必要だ。

³ 5.3 氷造形の初期実験

⁴ 水が綺麗な線が引けて凍るか、積層されるか、開発した氷をマテリアルとした3Dプリンターが
⁵ 実際に氷の造形ができるかをテストした。線を引く実験では、水と砂糖の割合が1:1のものを使
⁶ い、印刷速度、押し出し量、ライン幅、の調整を行った。純粋な水に比べ、砂糖を混ぜた水は凍
⁷ るまでの速度が遅く、速度を上げすぎると、凍る前に次の層の造形が始まってしまう一方、遅す
⁸ ぎると、ノズルと造形物が凍ってしまい造形ができなくなることが分かった。押し出し量、ライ
⁹ ン幅、については、ノズルの太さ(1mm)を踏まえたうえで、調整しないと、スカスカの造形ま
¹⁰ たは、水が飛び出したような造形になってしまった事が分かった。この時の最適なパラメーターは、
¹¹ ライン幅:3.0mm 印刷速度:50.0mm ということが分かった。このパラメータを基に素材ごとの
¹² 積層の実験を行った。

¹³ 5.4 氷の積層実験

¹⁴ 初期実験から Slic3r のパラメータを次のように設定した。ノズルの移動速度のパラメータであ
¹⁵ る Speed を 50mm/s に設定し、次のレイヤーを造形する際ノズルをどれだけ上げるかを決める
¹⁶ パラメータの Layer height を 0.1mm、外壁の厚みの層数を設定するパラメーターの Perimeters
¹⁷ を 35 に変更した。⁶ この設定で図 7.1 のような GCode を生成した。高さが 5mm に設定され
¹⁸ ており、5mm の高さまで積層できるかを調査した。⁸ 造形の際に、水の押し出し量が少なすぎ
¹⁹ ると、特定の場所だけ積層され一部が全く造形されないと言う結果になった。水の押し出し量
²⁰ は、フィラメントの押し出し量を設定する、Extrusion multiplier のパラメータで制御できる。
²¹ 初期は 0.1 で始め 0.1 ずつ足していく造形した。結果 ¹¹ Extrusion multiplier が 0.4 の時にう
²² まく積層ができ、図 7.2 のような氷を造形することができた。¹² 飛び出している部分は最初の造

1 形の際に生まれるもので、今回は無視する。計測の結果、高 13 さ 5mm、幅 4mm と言う結果に
2 なった。14 印刷中にヘッドパーツが造形中の氷に干渉する場面があったがヘッドが加熱されてい
3 たため、15 溶かしながら進み造形の失敗を防いでいた。また、押し出される水は少量でアルミト
4 レーや氷に 16 表面張力で吸い付き、凍ることで積層されていることがわかった。ヘッドの上げ率
5 や温度を調節 17 することで、さらに精度や効率を上げることができる可能性がある。18 押し出
6 し量が多すぎると、図 7.3 のようにところどころふくらみのある形状になってしまう。

1

第 6 章

2

今後の展望

3

6.1 段落と改行

謝辞

2 本論文の制作にあたり、指導教員である羽田久一教授には多大な助言、指導を受け賜りました。

3 深く感謝を申し上げます。AED.lab の皆さんも様々な助言やサポートをして頂いたこに感謝の意

4 を表します。皆さんのご支援やご意見は、本研究において、ひいては、自分の大きな糧となり、完

5 成に欠かせないものであったことをここで述べさせていただきます。重ねて心からの感謝の意を

6 込めて、筆を置かせて頂きます。ご清覧いただきありがとうございました。

参考文献

1

- 2 [1] 岡田耕治, 渡邊洋輔, 齊藤梓, 川上勝, 古川英光. 3d ゲルプリンティング. ネットワークポリ
3 マー論文集, Vol. 37, No. 2, pp. 81–87, 2016.
- 4 [2] Suntory-3D on the Rocks. 2017-10-16.
- 5 [3] Huaishu Peng, Jennifer Mankoff, Scott E. Hudson, and James McCann. A layered fabric
6 3d printer for soft interactive objects. In *Proceedings of the 33rd Annual ACM Conference*
7 on *Human Factors in Computing Systems*, Vol. 2, pp. 1789–1798, 2015.
- 8 [4] Michael Stern, Giorgia Franchin, Markus Kayser, John Klein, Chikara Inamura, Shreya-
9 Dave, James C Weaver, Peter Houk, Paolo Colombo, Maria Yang, and Neri Oxman.
10 Additive manufacturing of optically transparent glass. *3D Printing and Additive Manu*
11 *facturing*, Vol. 2, No. 3, pp. 92–105, 2015.
- 12 [5] 鈴木祐哉, 高岸賢輔, 梅津信二郎. 静電インクジェット方式 3d プリンタによる後年度食品材
13 料の高精度プリント. ライフサポート学会, 2016.
- 14 [6] 八角邦夫. 静電インクジェット方式 3d プリンタによる後年度食品材料の高精度プリント. 科
15 学と教育, Vol. 68, No. 2, pp. 66–67, 2020.
- 16 [7] 陶山剛. 3d プリンタのセラミックスへの適用. ニューセラミックス懇話会, Vol. 65, pp. 3–8,
17 2018.
- 18 [8] 藤田大樹, 中野亜希人, 羽田久一. Elsa:氷を素材とした 3d プリンターの開発. 研究報告デジ
19 タルコンテンツクリエーション (DCC) , Vol. 2017, No. DCC-17, pp. 1–7, 2017.
- 20 [9] 藤田大樹, 中野亜希人, 羽田久一. Elsa:氷を素材とした 3d プリンターの開発. 研究報告デジ
21 タルコンテンツクリエーション (DCC) , Vol. 2017, No. DCC-17, pp. 1–7, 2017.
- 22 [10] 谷貴幸, 後藤啓光. 放電現象を利用したインクジェット型金属 3d プリンター開発に関する基

1 硏究. 筑波技術大学テクノレポート, Vol. 25, No. 2, pp. 77–79, 2018.
2 本研究のために制作した 3Dprinter のファームウェア. Marlin 内にあるファイル名 Configu-
3 ration.h のコードである.

```
4  
5  
6  
7     /**  
8     * Marlin 3D Printer Firmware  
9     * Copyright (C) 2016 MarlinFirmware [ https://github.com/MarlinFirmware/Ma  
10    *  
11    * Based on Sprinter and grbl.  
12    * Copyright (C) 2011 Camiel Gubbels / Erik van der Zalm  
13    *  
14    * This program is free software: you can redistribute it and/or modify  
15    * it under the terms of the GNU General Public License as published by  
16    * the Free Software Foundation, either version 3 of the License, or  
17    * (at your option) any later version.  
18    *  
19    * This program is distributed in the hope that it will be useful,  
20    * but WITHOUT ANY WARRANTY; without even the implied warranty of  
21    * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
22    * GNU General Public License for more details.  
23    *  
24    * You should have received a copy of the GNU General Public License  
25    * along with this program. If not, see <http://www.gnu.org/licenses/>.  
26    *  
27    */  
28  
29    /**  
30    * Configuration.h
```

```

1   *
2   * Basic settings such as:
3   *
4   * - Type of electronics
5   * - Type of temperature sensor
6   * - Printer geometry
7   * - Endstop configuration
8   * - LCD controller
9   * - Extra features
10  *
11  * Advanced settings can be found in Configuration_adv.h
12  *
13  */
14 #ifndef CONFIGURATION_H
15 #define CONFIGURATION_H
16 #define CONFIGURATION_H_VERSION 010109
17
18 //=====
19 //===== Getting Started =====
20 //=====
21
22 /**
23  * Here are some standard links for getting your machine calibrated:
24  *
25  * http://reprap.org/wiki/Calibration
26  * http://youtu.be/wAL9d7FgInk
27  * http://calculator.josefprusa.cz
28  * http://reprap.org/wiki/Triffid_Hunter%27s_Calibration_Guide
29  * http://www.thingiverse.com/thing:5573
30  * https://sites.google.com/site/repraplogphase/calibration-of-your-reprap
31  * http://www.thingiverse.com/thing:298812
32 */

```

```

1
2 //=====
3 //===== DELTA Printer =====
4 //=====
5 // For a Delta printer start with one of the configuration files in the
6 // example_configurations/delta directory and customize for your machine.
7 //
8
9 //=====
10 //===== SCARA Printer =====
11 //=====
12 // For a SCARA printer start with the configuration files in
13 // example_configurations/SCARA and customize for your machine.
14 //
15
16 //=====
17 //===== HANGPRINTER =====
18 //=====
19 // For a Hangprinter start with the configuration file in the
20 // example_configurations/hangprinter directory and customize for your mac
21 //
22
23 // @section info
24
25 // User-specified version info of this build to display in [Prontorface , etc]
26 // startup. Implementation of an idea by Prof Braino to inform user that an
27 // build by the user have been successfully uploaded into firmware.
28 #define STRING_CONFIG_H_AUTHOR "(davidramiro)" // Who made the changes.
29 #define SHOW_BOOTSCREEN
30 #define STRING_SPLASH_LINE1 SHORT_BUILD_VERSION // will be shown during boot
31 #define STRING_SPLASH_LINE2 CUSTOM_BUILD_VERSION // will be shown during boot
32

```

```

1  /**
2   * *** VENDORS PLEASE READ ***
3   *
4   * Marlin allows you to add a custom boot image for Graphical LCDs.
5   * With this option Marlin will first show your custom screen followed
6   * by the standard Marlin logo with version number and web URL.
7   *
8   * We encourage you to take advantage of this new feature and we also
9   * respectfully request that you retain the unmodified Marlin boot screen.
10 */
11
12 // Enable to show the bitmap in Marlin/_Bootscreen.h on startup .
13 // #define SHOW_CUSTOM_BOOTSCREEN
14
15 // Enable to show the bitmap in Marlin/_Statusscreen.h on the status screen
16 // #define CUSTOM_STATUS_SCREEN_IMAGE
17
18 // @section machine
19
20 /**
21  * Select the serial port on the board to use for communication with the host
22  * This allows the connection of wireless adapters (for instance) to non-dedicated
23  * Serial port 0 is always used by the Arduino bootloader regardless of this setting
24  *
25  * :[0, 1, 2, 3, 4, 5, 6, 7]
26 */
27 #define SERIAL_PORT 0
28
29 /**
30  * This setting determines the communication speed of the printer .
31  *
32  * 250000 works in most cases , but you might try a lower speed if

```

```

1   * you commonly experience drop-outs during host printing .
2   * You may try up to 1000000 to speed up SD file transfer .
3   *
4   * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
5   */
6 #define BAUDRATE 250000
7
8 // Enable the Bluetooth serial interface on AT90USB devices
9 // #define BLUETOOTH
10
11 // The following define selects which electronics board you have .
12 // Please choose the name from boards.h that matches your setup
13 #ifndef MOTHERBOARD
14     #define MOTHERBOARD BOARD_TRIGORILLA_14
15 #endif
16
17 // Optional custom name for your RepStrap or other custom machine
18 // Displayed in the LCD "Ready" message
19 // #define CUSTOM_MACHINE_NAME "3D Printer"
20
21 // Define this to set a unique identifier for this printer , (Used by some P
22 // You can use an online service to generate a random UUID . (eg http://www.
23 // #define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
24
25 // @section extruder
26
27 // This defines the number of extruders
28 // :[1, 2, 3, 4, 5]
29 #define EXTRUDERS 1
30
31 // Generally expected filament diameter (1.75, 2.85, 3.0, ...). Used for V
32 #define DEFAULT_NOMINAL_FILAMENT_DIA 1.75

```

```

1
2 // For Cyclops or any "multi-extruder" that shares a single nozzle .
3 // #define SINGLENOZZLE
4
5 /**
6 * Prusa MK2 Single Nozzle Multi-Material Multiplexer , and variants .
7 *
8 * This device allows one stepper driver on a control board to drive
9 * two to eight stepper motors , one at a time , in a manner suitable
10 * for extruders .
11 *
12 * This option only allows the multiplexer to switch on tool-change .
13 * Additional options to configure custom E moves are pending .
14 */
15 // #define MK2MULTIPLEXER
16 #if ENABLED(MK2MULTIPLEXER)
17 // Override the default DIO selector pins here , if needed .
18 // Some pins files may provide defaults for these pins .
19 // #define E_MUX0_PIN 40 // Always Required
20 // #define E_MUX1_PIN 42 // Needed for 3 to 8 steppers
21 // #define E_MUX2_PIN 44 // Needed for 5 to 8 steppers
22 #endif
23
24 // A dual extruder that uses a single stepper motor
25 // #define SWITCHING_EXTRUDER
26 #if ENABLED(SWITCHING_EXTRUDER)
27 #define SWITCHING_EXTRUDER_SERVO_NR 0
28 #define SWITCHING_EXTRUDER_SERVO_ANGLES { 0 , 90 } // Angles for E0 , E1 [ ,
29 #if EXTRUDERS > 3
30 #define SWITCHING_EXTRUDER_E23_SERVO_NR 1
31 #endif
32 #endif

```

```

1
2 // A dual-nozzle that uses a servomotor to raise/lower one of the nozzles
3 // #define SWITCHING_NOZZLE
4 #if ENABLED(SWITCHING_NOZZLE)
5   #define SWITCHING_NOZZLE_SERVO_NR 0
6   #define SWITCHING_NOZZLE_SERVO_ANGLES { 0, 90 }      // Angles for E0, E1
7   // #define HOTEND_OFFSET_Z { 0.0, 0.0 }
8 #endif
9
10 /**
11 * Two separate X-carriages with extruders that connect to a moving part
12 * via a magnetic docking mechanism. Requires SOL1_PIN and SOL2_PIN.
13 */
14 // #define PARKING_EXTRUDER
15 #if ENABLED(PARKING_EXTRUDER)
16   #define PARKING_EXTRUDER_SOLENOIDS_INVERT           // If enabled, the so
17   #define PARKING_EXTRUDER_SOLENOIDS_PINS_ACTIVE LOW // LOW or HIGH pin si
18   #define PARKING_EXTRUDER_SOLENOIDS_DELAY 250        // Delay (ms) for ma
19   #define PARKING_EXTRUDER_PARKING_X { -78, 184 }     // X positions for pa
20   #define PARKING_EXTRUDER_GRAB_DISTANCE 1            // mm to move beyond
21   #define PARKING_EXTRUDER_SECURITY_RAISE 5          // Z-raise before pa
22   #define HOTEND_OFFSET_Z { 0.0, 1.3 }                // Z-offsets of the t
23#endif
24
25 /**
26 * "Mixing Extruder"
27 * — Adds G-codes M163 and M164 to set and "commit" the current mix facto
28 * — Extends the stepping routines to move multiple steppers in proportion
29 * — Optional support for Repetier Firmware's 'M164 S<index>' supporting
30 * — This implementation supports up to two mixing extruders.
31 * — Enable DIRECT_MIXING_IN_G1 for M165 and mixing in G1 (from Pia Taub
32 */

```

```

1 // #define MIXING_EXTRUDER
2 #if ENABLED(MIXING_EXTRUDER)
3     #define MIXING_STEPPERS 2          // Number of steppers in your mixing ext
4     #define MIXING_VIRTUAL_TOOLS 16    // Use the Virtual Tool method with M163
5     // #define DIRECT_MIXING_IN_G1    // Allow ABCDHII mix factors in G1 movem
6 #endif
7
8 // Offset of the extruders (uncomment if using more than one and relying on
9 // The offset has to be X=0, Y=0 for the extruder 0 hotend (default extrude
10 // For the other hotends it is their distance from the extruder 0 hotend.
11 // #define HOTEND_OFFSET_X {0.0, 20.00} // (in mm) for each extruder , offset
12 // #define HOTEND_OFFSET_Y {0.0, 5.00}  // (in mm) for each extruder , offset
13
14 // @section machine
15
16 /**
17 * Select your power supply here. Use 0 if you haven't connected the PS_ON.
18 *
19 * 0 = No Power Switch
20 * 1 = ATX
21 * 2 = X-Box 360 203Watts (the blue wire connected to PS_ON and the red wire
22 *
23 * :{ 0:'No power switch' , 1:'ATX' , 2:'X-Box 360' }
24 */
25 #define POWERSUPPLY 0
26
27 #if POWERSUPPLY > 0
28     // Enable this option to leave the PSU off at startup.
29     // Power to steppers and heaters will need to be turned on with M80.
30     // #define PS_DEFAULT_OFF
31
32 // #define AUTOPOWERCONTROL           // Enable automatic control of the PSU

```

```

1 #if ENABLED(AUTO_POWER_CONTROL)
2     #define AUTO_POWER_FANS           // Turn on PSU if fans need power
3     #define AUTO_POWER_E_FANS
4     #define AUTO_POWER_CONTROLLERFAN
5     #define POWER_TIMEOUT 30
6 #endif
7
8#endif
9
10// @section temperature
11
12//=====
13//===== Thermal Settings =====
14//=====
15
16/***
17 * —NORMAL IS 4.7kohm PULLUP!— 1kohm pullup can be used on hotend sensor,
18 *
19 * Temperature sensors available:
20 *
21 * -4 : thermocouple with AD8495
22 * -3 : thermocouple with MAX31855 (only for sensor 0)
23 * -2 : thermocouple with MAX6675 (only for sensor 0)
24 * -1 : thermocouple with AD595
25 * 0 : not used
26 * 1 : 100k thermistor – best choice for EPCOS 100k (4.7k pullup)
27 * 2 : 200k thermistor – ATC Semitec 204GT-2 (4.7k pullup)
28 * 3 : Mendel-parts thermistor (4.7k pullup)
29 * 4 : 10k thermistor !! do not use it for a hotend. It gives bad resol
30 * 5 : 100K thermistor – ATC Semitec 104GT-2/104NT-4-R025H42G (Used in
31 * 501 : 100K Zonestar (Tronxy X3A) Thermistor
32 * 6 : 100k EPCOS – Not as accurate as table 1 (created using a fluke t

```

```

1   *      7 : 100k Honeywell thermistor 135-104LAG-J01 (4.7k pullup)
2   *      71 : 100k Honeywell thermistor 135-104LAF-J01 (4.7k pullup)
3   *      8 : 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup)
4   *      9 : 100k GE Sensing AL03006-58.2K-97-G1 (4.7k pullup)
5   *     10 : 100k RS thermistor 198-961 (4.7k pullup)
6   *     11 : 100k beta 3950 1% thermistor (4.7k pullup)
7   *     12 : 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup) (calibrated)
8   *     13 : 100k Hisens 3950 1% up to ° 300C for hotend "Simple ONE" & "Hotend"
9   *     15 : 100k thermistor calibration for JGAurora A5 hotend
10  *    20 : the PT100 circuit found in the Ultimainboard V2.x
11  *    60 : 100k Maker's Tool Works Kapton Bed Thermistor beta=3950
12  *    66 : 4.7M High Temperature thermistor from Dyze Design
13  *    70 : the 100K thermistor found in the bq Hephestos 2
14  *    75 : 100k Generic Silicon Heat Pad with NTC 100K MGB18-104F39050L32 t
15  *
16  *      1k ohm pullup tables - This is atypical, and requires changing out
17  *                                (but gives greater accuracy and more stable)
18  *    51 : 100k thermistor - EPCOS (1k pullup)
19  *    52 : 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
20  *    55 : 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head)
21  *
22  *    1047 : Pt1000 with 4k7 pullup
23  *    1010 : Pt1000 with 1k pullup (non standard)
24  *    147 : Pt100 with 4k7 pullup
25  *    110 : Pt100 with 1k pullup (non standard)
26  *
27  *      Use these for Testing or Development purposes. NEVER for product
28  *    998 : Dummy Table that ALWAYS reads ° 25C or the temperature defined by
29  *    999 : Dummy Table that ALWAYS reads ° 100C or the temperature defined by
30  *
31  * :{ '0': "Not used", '1':"100k / 4.7k - EPCOS", '2':"200k / 4.7k - ATC Semitec
32  1% up to ° 300C for hotend 'Simple ONE' & hotend 'All In ONE', '20':"PT100

```

```

1   */
2 #define TEMP_SENSOR_0 5
3 #define TEMP_SENSOR_1 0
4 #define TEMP_SENSOR_2 0
5 #define TEMP_SENSOR_3 0
6 #define TEMP_SENSOR_4 0
7 #define TEMP_SENSOR_BED 1
8 #define TEMP_SENSOR_CHAMBER 0
9
10 // Dummy thermistor constant temperature readings , for use with 998 and 999
11 #define DUMMY_THERMISTOR_998_VALUE 25
12 #define DUMMY_THERMISTOR_999_VALUE 100
13
14 // Use temp sensor 1 as a redundant sensor with sensor 0. If the readings
15 // from the two sensors differ too much the print will be aborted .
16 // #define TEMP_SENSOR_1_AS_REDUNDANT
17 #define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10
18
19 // Extruder temperature must be close to target for this long before M109
20 #define TEMP_RESIDENCY_TIME 10 // (seconds)
21 #define TEMP_HYSTERESIS 3 // (degC) range of +/- temperatures considered
22 #define TEMP_WINDOW 1 // (degC) Window around target to start the
23
24 // Bed temperature must be close to target for this long before M190 returns
25 #define TEMP_BED_RESIDENCY_TIME 10 // (seconds)
26 #define TEMP_BED_HYSTERESIS 3 // (degC) range of +/- temperatures considered
27 #define TEMP_BED_WINDOW 1 // (degC) Window around target to start the
28
29 // The minimal temperature defines the temperature below which the heater
30 // to check that the wiring to the thermistor is not broken .
31 // Otherwise this would lead to the heater being powered on all the time .
32 #define HEATER_0_MINTEMP 5

```

```

1 #define HEATER1_MINTEMP 5
2 #define HEATER2_MINTEMP 5
3 #define HEATER3_MINTEMP 5
4 #define HEATER4_MINTEMP 5
5 #define BED_MINTEMP 5
6
7 // When temperature exceeds max temp, your heater will be switched off.
8 // This feature exists to protect your hotend from overheating accidentally.
9 // You should use MINTEMP for thermistor short/failure protection.
10 #define HEATER0_MAXTEMP 285
11 #define HEATER1_MAXTEMP 275
12 #define HEATER2_MAXTEMP 275
13 #define HEATER3_MAXTEMP 275
14 #define HEATER4_MAXTEMP 275
15 #define BED_MAXTEMP 135
16
17 //=====
18 //===== PID Settings =====
19 //=====
20 // PID Tuning Guide here: http://reprap.org/wiki/PID\_Tuning
21
22 // Comment the following line to disable PID and enable bang-bang.
23 #define PIDTEMP
24 #define BANGMAX 255      // Limits current to nozzle while in bang-bang mode
25 #define PID_MAX_BANGMAX // Limits current to nozzle while PID is active (so it can't
26 #define PID_K1 0.95       // Smoothing factor within any PID loop
27 #if ENABLED(PIDTEMP)
28     //#define PID_AUTOTUNE_MENU // Add PID Autotune to the LCD "Temperature"
29     //#define PID_DEBUG // Sends debug data to the serial port.
30     //#define PID_OPENLOOP 1 // Puts PID in open loop. M104/M140 sets the output
31     //#define SLOW_PWMHEATERS // PWM with very low frequency (roughly 0.125Hz)
32     //#define PID_PARAMS_PER_HOTEND // Uses separate PID parameters for each

```

```

1 // Set/get with gcode: M301 E[ extruder number ] S[ set value ] R[ read value ]
2 #define PID_FUNCTIONAL_RANGE 10 // If the temperature difference between
3 // is more than PID_FUNCTIONAL_RANGE then
4
5 // If you are using a pre-configured hotend then you can use one of the
6
7 // i3 Mega stock v5 hotend , 40W heater cartridge Ω(3.6 @ ° 22C)
8 #define DEFAULT_Kp 15.94
9 #define DEFAULT_Ki 1.17
10 #define DEFAULT_Kd 54.19
11
12 // Ultimaker
13 // #define DEFAULT_Kp 22.2
14 // #define DEFAULT_Ki 1.08
15 // #define DEFAULT_Kd 114
16
17 // MakerGear
18 // #define DEFAULT_Kp 7.0
19 // #define DEFAULT_Ki 0.1
20 // #define DEFAULT_Kd 12
21
22 // Mendel Parts V9 on 12V
23 // #define DEFAULT_Kp 63.0
24 // #define DEFAULT_Ki 2.25
25 // #define DEFAULT_Kd 440
26
27 #endif // PIDTEMP
28
29 //=====
30 //===== PID > Bed Temperature Control =====
31 //=====
32

```

```

1  /**
2   * PID Bed Heating
3   *
4   * If this option is enabled set PID constants below.
5   * If this option is disabled , bang–bang will be used and BED_LIMIT_SWITCHING
6   *
7   * The PID frequency will be the same as the extruder PWM.
8   * If PID_dT is the default , and correct for the hardware/configuration , th
9   * which is fine for driving a square wave into a resistive load and does n
10  * impact FET heating. This also works fine on a Fotek SSR–10DA Solid State
11  * heater. If your configuration is significantly different than this and y
12  * the issues involved , don't use bed PID until someone else verifies that
13  */
14 #define PIDTEMPBED
15
16 #define MAX_CYCLE_TIME_PID_AUTOTUNE 40L
17
18 //#define BED_LIMIT_SWITCHING
19
20 /**
21  * Max Bed Power
22  * Applies to all forms of bed control (PID, bang–bang , and bang–bang with
23  * When set to any value below 255, enables a form of PWM to the bed that a
24  * so don't use it unless you are OK with PWM on your bed. (See the comment
25  */
26 #define MAXBEDPOWER 255 // limits duty cycle to bed; 255=full current
27
28 #if ENABLED(PIDTEMPBED)
29
30 //#define PID_BED_DEBUG // Sends debug data to the serial port .
31
32 //Anycubic i3 Mega Ultrabase Ω(0.9 @ ° 22C)

```

```

1 #define DEFAULT_bedKp 251.78
2 #define DEFAULT_bedKi 49.57
3 #define DEFAULT_bedKd 319.73
4
5 //120V 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)
6 //from pidautotune
7 //##define DEFAULT_bedKp 97.1
8 //##define DEFAULT_bedKi 1.41
9 //##define DEFAULT_bedKd 1675.16
10
11 // FIND YOUR OWN: "M303 E-1 C8 S90" to run autotune on the bed at 90 deg
12 #endif // PIDTEMPBED
13
14 // @section extruder
15
16 /**
17 * Prevent extrusion if the temperature is below EXTRUDE_MINTEMP.
18 * Add M302 to set the minimum extrusion temperature and/or turn
19 * cold extrusion prevention on and off.
20 *
21 * *** IT IS HIGHLY RECOMMENDED TO LEAVE THIS OPTION ENABLED! ***
22 */
23 #define PREVENT_COLD_EXTRUSION変えた
24 //
25 #define EXTRUDE_MINTEMP -5
26
27 /**
28 * Prevent a single extrusion longer than EXTRUDE_MAXLENGTH.
29 * Note: For Bowden Extruders make this large enough to allow load/unload.
30 */
31 #define PREVENT_LENGTHY_EXTRUDE
32 #define EXTRUDE_MAXLENGTH 600

```

```

1
2 //=====
3 //===== Thermal Runaway Protection =====
4 //=====

5
6 /**
7 * Thermal Protection provides additional protection to your printer from
8 * and fire. Marlin always includes safe min and max temperature ranges which
9 * protect against a broken or disconnected thermistor wire.
10 *
11 * The issue: If a thermistor falls out, it will report the much lower
12 * temperature of the air in the room, and the the firmware will keep
13 * the heater on.
14 *
15 * If you get "Thermal Runaway" or "Heating failed" errors the
16 * details can be tuned in Configuration_adv.h
17 */
18
19 #define THERMAL_PROTECTION_HOTENDS // Enable thermal protection for all extruders
20 #define THERMAL_PROTECTION_BED // Enable thermal protection for the build plate
21
22 //=====
23 //===== Mechanical Settings =====
24 //=====

25
26 // @section machine
27
28 // Uncomment one of these options to enable CoreXY, CoreXZ, or CoreYZ kinematics
29 // either in the usual order or reversed
30 // #define COREXY
31 // #define COREXZ
32 // #define COREYZ

```

```

1 // #define COREYX
2 // #define COREZX
3 // #define COREZY
4
5 //=====
6 //===== Endstop Settings =====
7 //=====
8
9 // @section homing
10
11 // Specify here all the endstop connectors that are connected to any endstop
12 // Almost all printers will be using one per axis. Probes will use one or more
13 // extra connectors. Leave undefined any used for non-endstop and non-probe
14 #define USE_XMIN_PLUG
15 #define USE_YMIN_PLUG
16 #define USE_ZMIN_PLUG
17 #define USE_XMAX_PLUG
18 // #define USE_YMAX_PLUG
19 #define USE_ZMAX_PLUG
20
21 // Enable pullup for all endstops to prevent a floating state
22 #define ENDSTOPPULLUPS
23 #if DISABLED(ENDSTOPPULLUPS)
24     // Disable ENDSTOPPULLUPS to set pullups individually
25     // #define ENDSTOPPULLUP_XMAX
26     // #define ENDSTOPPULLUP_YMAX
27     // #define ENDSTOPPULLUP_ZMAX
28     // #define ENDSTOPPULLUP_XMIN
29     // #define ENDSTOPPULLUP_YMIN
30     // #define ENDSTOPPULLUP_ZMIN
31     // #define ENDSTOPPULLUP_ZMIN_PROBE
32 #endif

```

```

1
2 // Mechanical endstop with COM to ground and NC to Signal uses "false" here
3 #define X_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of
4 #define Y_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of
5 #define Z_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of
6 #define X_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of
7 #define Y_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of
8 #define Z_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of
9 #define Z_MIN_PROBE_ENDSTOP_INVERTING true // set to true to invert the logi

10 /**
11  * Stepper Drivers
12  *
13  *
14  * These settings allow Marlin to tune stepper driver timing and enable adv
15  * stepper drivers that support them. You may also override timing options
16  *
17  * A4988 is assumed for unspecified drivers.
18  *
19  * Options: A4988, DRV8825, LV8729, L6470, TB6560, TB6600, TMC2100,
20  *          TMC2130, TMC2130_STANDALONE, TMC2208, TMC2208_STANDALONE,
21  *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
22  *          TMC5130, TMC5130_STANDALONE
23  * :[ 'A4988', 'DRV8825', 'LV8729', 'L6470', 'TB6560', 'TB6600', 'TMC2100',
24  */
25 #define X_DRIVER_TYPE A4988 // comment out for stock drivers
26 #define Y_DRIVER_TYPE A4998 // comment out for stock drivers
27 #define Z_DRIVER_TYPE A4998 // comment out for stock drivers
28 #define X2_DRIVER_TYPE A4998
29 #define Y2_DRIVER_TYPE A4998
30 #define Z2_DRIVER_TYPE A4998 // comment out for stock drivers
31 #define E0_DRIVER_TYPE A4998 // comment out for stock drivers
32 #define E1_DRIVER_TYPE A4998 // comment out for stock drivers

```

```

1 #define E2_DRIVER_TYPE A4998
2 #define E3_DRIVER_TYPE A4998
3 #define E4_DRIVER_TYPE A4998
4
5 // Enable this feature if all enabled endstop pins are interrupt-capable.
6 // This will remove the need to poll the interrupt pins , saving many CPU cy
7 // #define ENDSTOP_INTERRUPTS_FEATURE
8
9 /**
10 * Endstop Noise Filter
11 *
12 * Enable this option if endstops falsely trigger due to noise.
13 * NOTE: Enabling this feature means adds an error of +/-0.2mm, so homing
14 * will end up at a slightly different position on each G28. This will also
15 * reduce accuracy of some bed probes.
16 * For mechanical switches , the better approach to reduce noise is to insta
17 * a 100 nanofarads ceramic capacitor in parallel with the switch , making i
18 * essentially noise-proof without sacrificing accuracy .
19 * This option also increases MCU load when endstops or the probe are enable
20 * So this is not recommended. USE AT YOUR OWN RISK.
21 * (This feature is not required for common micro-switches mounted on PCBs
22 * based on the Makerbot design , since they already include the 100nF capa
23 */
24 #define ENDSTOP_NOISE_FILTER
25
26 //=====
27 //===== Movement Settings =====
28 //=====
29 // @section motion
30
31 /**
32 * Default Settings

```

```

1   *
2   * These settings can be reset by M502
3   *
4   * Note that if EEPROM is enabled, saved values will override these.
5   */
6
7  /**
8   * With this option each E stepper can have its own factors for the
9   * following movement settings. If fewer factors are given than the
10  * total number of extruders, the last value applies to the rest.
11  */
12 // #define DISTINCT_E_FACTORS
13
14 /**
15  * Default Axis Steps Per Unit (steps/mm)
16  * Override with M92
17  *                                         X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
18 */
19
20 // ここちょい変えた。
21 #define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 400, 500 }
22
23 /**
24  * Default Max Feed Rate (mm/s)
25  * Override with M203
26  *                                         X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
27 */
28 #define DEFAULT_MAX_FEEDRATE { 500, 500, 6, 60 }
29
30 /**
31  * Default Max Acceleration (change/s) change = mm/s
32  * (Maximum start speed for accelerated moves)

```

```

1   * Override with M201
2   *
3   */
4 #define DEFAULT_MAX_ACCELERATION      { 3000, 2000, 60, 10000 }
5
6 /**
7  * Default Acceleration (change/s) change = mm/s
8  * Override with M204
9  *
10 * M204 P      Acceleration
11 * M204 R      Retract Acceleration
12 * M204 T      Travel Acceleration
13 */
14 #define DEFAULT_ACCELERATION          1500    // X, Y, Z and E acceleration
15 #define DEFAULT_RETRACT_ACCELERATION  3000    // E acceleration for retract
16 #define DEFAULT_TRAVEL_ACCELERATION   3000    // X, Y, Z acceleration for t
17
18 /**
19 * Default Jerk (mm/s)
20 * Override with M205 X Y Z E
21 *
22 * "Jerk" specifies the minimum speed change that requires acceleration .
23 * When changing speed and direction , if the difference is less than the
24 * value set here , it may happen instantaneously .
25 */
26 #define DEFAULT_XJERK                10.0
27 #define DEFAULT_YJERK                10.0
28 #define DEFAULT_ZJERK                0.4
29 #define DEFAULT_EJERK                5.0
30
31 /**
32 * S-Curve Acceleration

```

```

1   *
2   * This option eliminates vibration during printing by fitting a Bezier
3   * curve to move acceleration , producing much smoother direction changes.
4   *
5   * See https://github.com/synthetos/TinyG/wiki/Jerk-Controlled-Motion-Explained
6   */
7 #define S_CURVE_ACCELERATION

8
9 //=====
10 //===== Z Probe Options =====
11 //=====
12 // @section probes
13
14 //
15 // See http://marlinfw.org/docs/configuration/probes.html
16 //
17
18 /**
19 * Z_MIN_PROBEUSES_Z_MIN_ENDSTOP_PIN
20 *
21 * Enable this option for a probe connected to the Z Min endstop pin .
22 */
23 // #define Z_MIN_PROBEUSES_Z_MIN_ENDSTOP_PIN

24
25 /**
26 * Z_MIN_PROBE_ENDSTOP
27 *
28 * Enable this option for a probe connected to any pin except Z-Min .
29 * (By default Marlin assumes the Z-Max endstop pin .)
30 * To use a custom Z Probe pin , set Z_MIN_PROBE_PIN below .
31 *
32 * — The simplest option is to use a free endstop connector .

```

```

1   * — Use 5V for powered (usually inductive) sensors.
2   *
3   * — RAMPS 1.3/1.4 boards may use the 5V, GND, and Aux4→D32 pin:
4   *     — For simple switches connect...
5   *         — normally-closed switches to GND and D32.
6   *         — normally-open switches to 5V and D32.
7   *
8   * WARNING: Setting the wrong pin may have unexpected and potentially
9   * disastrous consequences. Use with caution and do your homework.
10  *
11  */
12 #define Z_MIN_PROBE_ENDSTOP
13
14 /**
15  * Probe Type
16  *
17  * Allen Key Probes, Servo Probes, Z-Sled Probes, FIX_MOUNTED_PROBE, etc.
18  * Activate one of these to use Auto Bed Leveling below.
19  */
20
21 /**
22  * The "Manual Probe" provides a means to do "Auto" Bed Leveling without a
23  * Use G29 repeatedly, adjusting the Z height at each point with movement or
24  * or (with LCD_BED_LEVELING) the LCD controller.
25  */
26 #define PROBEMANUALLY
27 // #define MANUAL_PROBE_START_Z 0.2
28
29 /**
30  * A Fix-Mounted Probe either doesn't deploy or needs manual deployment.
31  * (e.g., an inductive probe or a nozzle-based probe-switch.)
32  */

```

```

1 //#define FIX_MOUNTED_PROBE
2
3 /**
4 * Z Servo Probe, such as an endstop switch on a rotating arm.
5 */
6 //#define Z_PROBE_SERVO_NR 0      // Defaults to SERVO 0 connector.
7 //#define Z_SERVO_ANGLES {70,0}   // Z Servo Deploy and Stow angles
8
9 /**
10 * The BLTouch probe uses a Hall effect sensor and emulates a servo.
11 */
12 //#define BLTOUCH
13
14 /**
15 * Enable one or more of the following if probing seems unreliable.
16 * Heaters and/or fans can be disabled during probing to minimize electrical
17 * noise. A delay can also be added to allow noise and vibration to settle.
18 * These options are most useful for the BLTouch probe, but may also improve
19 * readings with inductive probes and piezo sensors.
20 */
21 //#define PROBING_HEATERS_OFF      // Turn heaters off when probing
22 #if ENABLED(PROBING_HEATERS_OFF)
23     //#define WAIT_FOR_BED_HEATER    // Wait for bed to heat back up between
24 #endif
25 //#define PROBING_FANS_OFF        // Turn fans off when probing
26 //#define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from trig-
27
28 // A probe that is deployed and stowed with a solenoid pin (SOL1_PIN)
29 //#define SOLENOID_PROBE
30
31 // A sled-mounted probe like those designed by Charles Bell.
32 //#define Z_PROBE_SLED

```

```

1 // #define SLED_DOCKING_OFFSET 5 // The extra distance the X axis must trav
2
3 //
4 // For Z_PROBE_ALLEN_KEY see the Delta example configurations .
5 //
6
7 /**
8 *      Z Probe to nozzle (X,Y) offset , relative to (0 , 0).
9 *      X and Y offsets must be integers .
10 *
11 *      In the following example the X and Y offsets are both positive :
12 *      #define X_PROBE_OFFSET_FROM_EXTRUDER 10
13 *      #define Y_PROBE_OFFSET_FROM_EXTRUDER 10
14 *
15 *          +--- BACK ---+
16 *          |           |
17 *          L | (+) P | R <-- probe (20,20)
18 *          E |           | I
19 *          F | (-) N (+) | G <-- nozzle (10,10)
20 *          T |           | H
21 *          | (-)       | T
22 *          |
23 *          O--- FRONT ---+
24 *          (0,0)
25 */
26 #define X_PROBE_OFFSET_FROM_EXTRUDER 10 // X offset : -left +right
27 [ of the nozzle ]
28 #define Y_PROBE_OFFSET_FROM_EXTRUDER 10 // Y offset : -front +behind [ the n
29 #define Z_PROBE_OFFSET_FROM_EXTRUDER 0 // Z offset : -below +above
30 [ the nozzle ]
31
32 // Certain types of probes need to stay away from edges

```

```

1 #define MIN_PROBE_EDGE 10
2
3 // X and Y axis travel speed (mm/m) between probes
4 #define XY_PROBE_SPEED 8000
5
6 // Feedrate (mm/m) for the first approach when double-probing (MULTIPLE_PROBING)
7 #define Z_PROBE_SPEED_FAST HOMING_FEEDRATE_Z
8
9 // Feedrate (mm/m) for the "accurate" probe of each point
10 #define Z_PROBE_SPEED_SLOW (Z_PROBE_SPEED_FAST / 2)
11
12 // The number of probes to perform at each point.
13 // Set to 2 for a fast/slow probe, using the second probe result.
14 // Set to 3 or more for slow probes, averaging the results.
15 // #define MULTIPLE_PROBING 2
16
17 /**
18 * Z probes require clearance when deploying, stowing, and moving between
19 * probe points to avoid hitting the bed and other hardware.
20 * Servo-mounted probes require extra space for the arm to rotate.
21 * Inductive probes need space to keep from triggering early.
22 *
23 * Use these settings to specify the distance (mm) to raise the probe (or
24 * lower the bed). The values set here apply over and above any (negative)
25 * probe Z Offset set with Z_PROBE_OFFSET_FROM_EXTRUDER, M851, or the LCD.
26 * Only integer values >= 1 are valid here.
27 *
28 * Example: 'M851 Z-5' with a CLEARANCE of 4 => 9mm from bed to nozzle.
29 * But: 'M851 Z+1' with a CLEARANCE of 2 => 2mm from bed to nozzle.
30 */
31 #define Z_CLEARANCE_DEPLOY_PROBE 10 // Z Clearance for Deploy/Stow
32 #define Z_CLEARANCE_BETWEEN_PROBES 5 // Z Clearance between probe points

```

```

1 #define Z_CLEARANCE_MULTI_probe      5 // Z Clearance between multiple probe
2 // #define Z_AFTER_PROBING          5 // Z position after probing is done
3
4 #define Z_PROBE_LOW_POINT          -2 // Farthest distance below the trigger
5
6 // For M851 give a range for adjusting the Z probe offset
7 #define Z_PROBE_OFFSET_RANGE_MIN -20
8 #define Z_PROBE_OFFSET_RANGE_MAX 20
9
10 // Enable the M48 repeatability test to test probe accuracy
11 // #define Z_MIN_PROBE_REPEATABILITY_TEST
12
13 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active
14 // :{ 0:'Low', 1:'High' }
15 #define X_ENABLE_ON 0
16 #define Y_ENABLE_ON 0
17 #define Z_ENABLE_ON 0
18 #define E_ENABLE_ON 0 // For all extruders
19
20 // Disables axis stepper immediately when it's not being used.
21 // WARNING: When motors turn off there is a chance of losing position accuracy
22 #define DISABLE_X false
23 #define DISABLE_Y false
24 #define DISABLE_Z false
25 // Warn on display about possibly reduced accuracy
26 // #define DISABLE_REDUCED_ACCURACY_WARNING
27
28 // @section extruder
29
30 #define DISABLE_E false // For all extruders
31 #define DISABLE_INACTIVE_EXTRUDER true // Keep only the active extruder enabled
32

```

```

1 // @section machine
2
3 // Invert the stepper direction. Change (or reverse the motor connector) if
4 #define INVERT_X_DIR true // set to true for stock drivers or TMC2208 with
5 #define INVERT_Y_DIR false // set to false for stock drivers or TMC2208 with
6 #define INVERT_Z_DIR false // set to false for stock drivers or TMC2208 with
7
8 // @section extruder
9
10 // For direct drive extruder v9 set to true, for geared extruder set to false
11 #define INVERT_E0_DIR false // set to false for stock drivers or TMC2208 with
12 #define INVERT_E1_DIR false // set to false for stock drivers or TMC2208 with
13 #define INVERT_E2_DIR false
14 #define INVERT_E3_DIR false
15 #define INVERT_E4_DIR false
16
17 // @section homing
18
19 // #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have
20
21 // #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unkn
22
23 // #define Z_HOME_HEIGHT 4 // (in mm) Minimal z height before homing (G28)
24 // Be sure you have this distance over your Z-
25
26 // Direction of endstops when homing; 1=MAX, -1=MIN
27 // :[-1,1]
28 #define X_HOME_DIR -1
29 #define Y_HOME_DIR -1
30 #define Z_HOME_DIR -1
31
32 // @section machine

```

```

1
2 // The size of the print bed
3 #define X_BED_SIZE 215
4 #define Y_BED_SIZE 215
5
6 // Travel limits (mm) after homing, corresponding to endstop positions.
7 #define X_MIN_POS -5
8 #define Y_MIN_POS 0
9 #define Z_MIN_POS 0
10 #define X_MAX_POS X_BED_SIZE
11 #define Y_MAX_POS Y_BED_SIZE
12 #define Z_MAX_POS 205
13
14 /**
15 * Software Endstops
16 *
17 * — Prevent moves outside the set machine bounds.
18 * — Individual axes can be disabled, if desired.
19 * — X and Y only apply to Cartesian robots.
20 * — Use 'M211' to set software endstops on/off or report current state
21 */
22
23 // Min software endstops constrain movement within minimum coordinate bound
24 #define MIN_SOFTWARE_ENDSTOPS
25 #if ENABLED(MIN_SOFTWARE_ENDSTOPS)
26 #define MIN_SOFTWARE_ENDSTOP_X
27 #define MIN_SOFTWARE_ENDSTOP_Y
28 #define MIN_SOFTWARE_ENDSTOP_Z
29#endif
30
31 // Max software endstops constrain movement within maximum coordinate bound
32 #define MAX_SOFTWARE_ENDSTOPS

```

```

1 #if ENABLED(MAX_SOFTWARE_ENDSTOPS)
2   #define MAX_SOFTWARE_ENDSTOP_X
3   #define MAX_SOFTWARE_ENDSTOP_Y
4   #define MAX_SOFTWARE_ENDSTOP_Z
5 #endif
6
7 #if ENABLED(MIN_SOFTWARE_ENDSTOPS) || ENABLED(MAX_SOFTWARE_ENDSTOPS)
8   // #define SOFT_ENDSTOPS_MENU_ITEM // Enable/Disable software endstops fr
9 #endif
10
11 /**
12  * Filament Runout Sensors
13  * Mechanical or opto endstops are used to check for the presence of filament
14  *
15  * RAMPS-based boards use SERVO3_PIN for the first runout sensor.
16  * For other boards you may need to define FIL_RUNOUT_PIN, FIL_RUNOUT2_PIN,
17  * By default the firmware assumes HIGH=FILAMENT PRESENT.
18 */
19 // #define FILAMENT_RUNOUT_SENSOR
20 #if ENABLED(FILAMENT_RUNOUT_SENSOR)
21   #define NUM_RUNOUT_SENSORS 1      // Number of sensors, up to one per extruder
22   #define FIL_RUNOUT_INVERTING false // set to true to invert the logic of the sensor
23   #define FIL_RUNOUT_PULLUP        // Use internal pullup for filament runout sensors
24   #define FILAMENT_RUNOUT_SCRIPT "M600"
25 #endif
26
27 //=====
28 //===== Bed Leveling =====
29 //=====
30 // @section calibrate
31
32 /**

```

```
1 * Choose one of the options below to enable G29 Bed Leveling. The parameters  
2 * and behavior of G29 will change depending on your selection.  
3 *  
4 * If using a Probe for Z Homing, enable Z_SAFE_HOMING also!  
5 *  
6 * — AUTO_BED_LEVELING_3POINT  
7 * Probe 3 arbitrary points on the bed (that aren't collinear)  
8 * You specify the XY coordinates of all 3 points.  
9 * The result is a single tilted plane. Best for a flat bed.  
10 *  
11 * — AUTO_BED_LEVELING_LINEAR  
12 * Probe several points in a grid.  
13 * You specify the rectangle and the density of sample points.  
14 * The result is a single tilted plane. Best for a flat bed.  
15 *  
16 * — AUTO_BED_LEVELING_BILINEAR  
17 * Probe several points in a grid.  
18 * You specify the rectangle and the density of sample points.  
19 * The result is a mesh, best for large or uneven beds.  
20 *  
21 * — AUTO_BED_LEVELING_UBL (Unified Bed Leveling)  
22 * A comprehensive bed leveling system combining the features and benefits  
23 * of other systems. UBL also includes integrated Mesh Generation, Mesh  
24 * Validation and Mesh Editing systems.  
25 *  
26 * — MESH_BED_LEVELING  
27 * Probe a grid manually  
28 * The result is a mesh, suitable for large or uneven beds. (See BILINEAR)  
29 * For machines without a probe, Mesh Bed Leveling provides a method to perform  
30 * leveling in steps so you can manually adjust the Z height at each grid point.  
31 * With an LCD controller the process is guided step-by-step.  
32 */
```

```

1 //#define AUTO_BED_LEVELING_3POINT
2 //#define AUTO_BED_LEVELING_LINEAR
3 //#define AUTO_BED_LEVELING_BILINEAR
4 //#define AUTO_BED_LEVELING_UBL
5 #define MESH_BED_LEVELING
6
7 /**
8  * Normally G28 leaves leveling disabled on completion. Enable
9  * this option to have G28 restore the prior leveling state.
10 */
11 //#define RESTORE_LEVELING_AFTER_G28
12
13 /**
14  * Enable detailed logging of G28, G29, M48, etc.
15  * Turn on with the command 'M111 S32'.
16  * NOTE: Requires a lot of PROGMEM!
17 */
18 //#define DEBUG_LEVELING_FEATURE
19
20 #if ENABLED(MESH_BED_LEVELING) || ENABLED(AUTO_BED_LEVELING_BILINEAR) || EN-
21 // Gradually reduce leveling correction until a set height is reached,
22 // at which point movement will be level to the machine's XY plane.
23 // The height can be set with M420 Z<height>
24 #define ENABLE_LEVELING_FADE_HEIGHT
25
26 // For Cartesian machines, instead of dividing moves on mesh boundaries,
27 // split up moves into short segments like a Delta. This follows the
28 // contours of the bed more closely than edge-to-edge straight moves.
29 #define SEGMENT_LEVELED_MOVES
30 #define LEVELED_SEGMENT_LENGTH 5.0 // (mm) Length of all segments (except
31
32 /**

```

```

1   * Enable the G26 Mesh Validation Pattern tool.
2   */
3 #define G26_MESH_VALIDATION
4 #if ENABLED(G26_MESH_VALIDATION)
5   #define MESH_TEST_NOZZLE_SIZE      0.4 // (mm) Diameter of primary nozzle
6   #define MESH_TEST_LAYER_HEIGHT    0.2 // (mm) Default layer height for
7   #define MESH_TEST_HOTEND_TEMP    200.0 // ° (C) Default nozzle temperature
8   #define MESH_TEST_BED_TEMP       60.0 // ° (C) Default bed temperature
9 #endif
10
11#endif
12
13#if ENABLED(AUTO_BED_LEVELING_LINEAR) || ENABLED(AUTO_BED_LEVELING_BILINEAR)
14
15  // Set the number of grid points per dimension.
16 #define GRID_MAX_POINTS_X 3
17 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
18
19  // Set the boundaries for probing (where the probe can reach).
20 #define LEFT_PROBE_BED_POSITION 25
21 #define RIGHT_PROBE_BED_POSITION 181
22 #define FRONT_PROBE_BED_POSITION 25
23 #define BACK_PROBE_BED_POSITION 185
24
25  // The Z probe minimum outer margin (to validate G29 parameters).
26 #define MIN_PROBE_EDGE 10
27
28  // Probe along the Y axis , advancing X after each column
29 // #define PROBE_Y_FIRST
30
31 #if ENABLED(AUTO_BED_LEVELING_BILINEAR)
32

```

```

1 // Beyond the probed grid , continue the implied tilt ?
2 // Default is to maintain the height of the nearest edge .
3 // #define EXTRAPOLATE_BEYOND_GRID
4
5 //
6 // Experimental Subdivision of the grid by Catmull-Rom method .
7 // Synthesizes intermediate points to produce a more detailed mesh .
8 //
9 // #define ABL_BILINEAR_SUBDIVISION
10 #if ENABLED(ABL_BILINEAR_SUBDIVISION)
11     // Number of subdivisions between probe points
12     #define BILINEAR_SUBDIVISIONS 3
13 #endif
14
15 #endif
16
17 #elif ENABLED(AUTO_BED_LEVELING_UBL)
18
19 //=====
20 //===== Unified Bed Leveling =====
21 //=====
22
23 // #define MESH_EDIT_GFX_OVERLAY // Display a graphics overlay while ed
24
25 #define MESH_INSET 1           // Set Mesh bounds as an inset region o
26 #define GRID_MAX_POINTS_X 10    // Don't use more than 15 points per ax
27 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
28
29 #define UBL_MESH_EDIT_MOVES_Z    // Sophisticated users prefer no movem
30 #define UBL_SAVE_ACTIVE_ON_M500   // Save the currently active mesh in th
31
32 // #define UBL_Z_RAISE_WHEN_OFF_MESH 2.5 // When the nozzle is off the mes

```

```

1 // as the Z-Height correction val
2
3 #elif ENABLED(MESH_BED_LEVELING)
4
5 //=====
6 //===== Mesh =====
7 //=====
8
9 #define MESH_INSET 10           // Set Mesh bounds as an inset region of t
10 #define GRID_MAX_POINTS_X 5    // Don't use more than 7 points per axis,
11 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
12
13 //#define MESH_G28_REST_ORIGIN // After homing all axes ('G28' or 'G28 XY'
14
15 #endif // BED_LEVELING
16
17 /**
18 * Points to probe for all 3-point Leveling procedures.
19 * Override if the automatically selected points are inadequate.
20 */
21 #if ENABLED(AUTO_BED_LEVELING_3POINT) || ENABLED(AUTO_BED_LEVELINGUBL)
22 //#define PROBE_PT_1_X 15
23 //#define PROBE_PT_1_Y 180
24 //#define PROBE_PT_2_X 15
25 //#define PROBE_PT_2_Y 20
26 //#define PROBE_PT_3_X 170
27 //#define PROBE_PT_3_Y 20
28#endif
29
30 /**
31 * Add a bed leveling sub-menu for ABL or MBL.
32 * Include a guided procedure if manual probing is enabled.

```

```

1   */
2 // #define LCD_BED_LEVELING
3
4 #if ENABLED(LCD_BED_LEVELING)
5   #define MBL_Z_STEP 0.025      // Step size while manually probing Z axis.
6   #define LCD_PROBEZ_RANGE 4 // Z Range centered on Z_MIN_POS for LCD Z axis
7 #endif
8
9 // Add a menu item to move between bed corners for manual bed adjustment
10 // #define LEVEL_BED_CORNERS
11
12 #if ENABLED(LEVEL_BED_CORNERS)
13   #define LEVEL_CORNERS_INSET 30      // (mm) An inset for corner leveling
14   #define LEVEL_CORNERS_Z_HOP 4.0    // (mm) Move nozzle up before moving bed
15   // #define LEVEL_CENTER_TOO        // Move to the center after the last corner
16 #endif
17
18 /**
19 * Commands to execute at the end of G29 probing.
20 * Useful to retract or move the Z probe out of the way.
21 */
22 // #define Z_PROBE_END_SCRIPT "G1 Z10 F12000\nG1 X15 Y330\nG1 Z0.5\nG1 Z10"
23
24
25 // @section homing
26
27 // The center of the bed is at (X=0, Y=0)
28 // #define BED_CENTER_AT_0_0
29
30 // Manually set the home position. Leave these undefined for automatic setting.
31 // For DELTA this is the top-center of the Cartesian print volume.
32 // #define MANUAL_X_HOME_POS 0

```

```

1 // #define MANUAL_Y_HOME_POS 0
2 // #define MANUAL_Z_HOME_POS 0
3
4 // Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
5 //
6 // With this feature enabled:
7 //
8 // - Allow Z homing only after X and Y homing AND stepper drivers still enable.
9 // - If stepper drivers time out, it will need X and Y homing again before Z homing.
10 // - Move the Z probe (or nozzle) to a defined XY point before Z Homing whenever.
11 // - Prevent Z homing when the Z probe is outside bed area.
12 //
13 // #define Z_SAFE_HOMING
14
15 #if ENABLED(Z_SAFE_HOMING)
16     #define Z_SAFE_HOMING_X_POINT ((X_BED_SIZE) / 2)      // X point for Z homing.
17     #define Z_SAFE_HOMING_Y_POINT ((Y_BED_SIZE) / 2)      // Y point for Z homing.
18 #endif
19
20 // Homing speeds (mm/m)
21 #define HOMING_FEEDRATE_XY (50*60)
22 #define HOMING_FEEDRATE_Z (4*60)
23
24 // @section calibrate
25
26 /**
27 * Bed Skew Compensation
28 *
29 * This feature corrects for misalignment in the XYZ axes.
30 *
31 * Take the following steps to get the bed skew in the XY plane:
32 *   1. Print a test square (e.g., https://www.thingiverse.com/thing:2563185)

```

```

1   * 2. For XY_DIAG_AC measure the diagonal A to C
2   * 3. For XY_DIAG_BD measure the diagonal B to D
3   * 4. For XY_SIDE_AD measure the edge A to D
4   *
5   * Marlin automatically computes skew factors from these measurements.
6   * Skew factors may also be computed and set manually:
7   *
8   * - Compute AB      : SQRT(2*AC*AC+2*BD*BD-4*AD*AD)/2
9   * - XY_SKEW_FACTOR : TAN(PI/2-ACOS((AC*AC-AB*AB-AD*AD)/(2*AB*AD)))
10  *
11  * If desired, follow the same procedure for XZ and YZ.
12  * Use these diagrams for reference:
13  *
14  *      Y           Z           Z
15  *      ^     B---C     ^     B---C     ^     B---C
16  *      |     /     /     |     /     /     |     /     /
17  *      |     /     /     |     /     /     |     /     /
18  *      |   A---D     |   A---D     |   A---D
19  *      +----->X     +----->X     +----->Y
20  *      XY_SKEW_FACTOR    XZ_SKEW_FACTOR    YZ_SKEW_FACTOR
21  */
22 // #define SKEW_CORRECTION
23
24 #if ENABLED(SKEW_CORRECTION)
25 // Input all length measurements here:
26 #define XY_DIAG_AC 282.8427124746
27 #define XY_DIAG_BD 282.8427124746
28 #define XY_SIDE_AD 200
29
30 // Or, set the default skew factors directly here
31 // to override the above measurements:
32 #define XY_SKEW_FACTOR 0.0

```

```

1
2 // #define SKEW_CORRECTION_FOR_Z
3 #if ENABLED(SKEW_CORRECTION_FOR_Z)
4     #define XZ_DIAG_AC 282.8427124746
5     #define XZ_DIAG_BD 282.8427124746
6     #define YZ_DIAG_AC 282.8427124746
7     #define YZ_DIAG_BD 282.8427124746
8     #define YZ_SIDE_AD 200
9     #define XZ_SKEW_FACTOR 0.0
10    #define YZ_SKEW_FACTOR 0.0
11 #endif
12
13 // Enable this option for M852 to set skew at runtime
14 // #define SKEW_CORRECTION_GCODE
15 #endif
16
17 //=====
18 //===== Additional Features =====
19 //=====
20
21 // @section extras
22
23 //
24 // EEPROM
25 //
26 // The microcontroller can store settings in the EEPROM, e.g. max velocity .
27 // M500 — stores parameters in EEPROM
28 // M501 — reads parameters from EEPROM (if you need reset them after you ch
29 // M502 — reverts to the default "factory settings". You still need to sto
30 //
31 #define EEPROMSETTINGS // Enable for M500 and M501 commands
32 // #define DISABLE_M503 // Saves ~2700 bytes of PROGMEM. Disable for rele

```

```

1 #define EEPROMCHITCHAT      // Give feedback on EEPROM commands. Disable to s
2
3 //
4 // Host Keepalive
5 //
6 // When enabled Marlin will send a busy status message to the host
7 // every couple of seconds when it can't accept commands.
8 //
9 #define HOST_KEEPALIVE_FEATURE           // Disable this if your host doesn't
10 #define DEFAULT_KEEPALIVE_INTERVAL 2    // Number of seconds between "busy" m
11 #define BUSY_WHILE_HEATING              // Some hosts require "busy" messages
12
13 //
14 // M100 Free Memory Watcher
15 //
16 // #define M100_FREE_MEMORY_WATCHER     // Add M100 (Free Memory Watcher) to
17
18 //
19 // G20/G21 Inch mode support
20 //
21 // #define INCH_MODE_SUPPORT
22
23 //
24 // M149 Set temperature units support
25 //
26 // #define TEMPERATURE_UNITS_SUPPORT
27
28 // @section temperature
29
30 // Preheat Constants
31 #define PREHEAT_1_TEMP_HOTEND 180
32 #define PREHEAT_1_TEMP_BED      70

```

```

1 #define PREHEAT_1_FAN_SPEED      0 // Value from 0 to 255
2
3 #define PREHEAT_2_TEMP_HOTEND 240
4 #define PREHEAT_2_TEMP_BED    110
5 #define PREHEAT_2_FAN_SPEED    0 // Value from 0 to 255
6
7 /**
8 * Nozzle Park
9 *
10 * Park the nozzle at the given XYZ position on idle or G27.
11 *
12 * The "P" parameter controls the action applied to the Z axis:
13 *
14 *   P0 (Default) If Z is below park Z raise the nozzle.
15 *   P1 Raise the nozzle always to Z-park height.
16 *   P2 Raise the nozzle by Z-park amount, limited to Z_MAX_POS.
17 */
18 #define NOZZLE_PARK_FEATURE
19
20 #if ENABLED(NOZZLE_PARK_FEATURE)
21     // Specify a park position as { X, Y, Z }
22     #define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }
23     #define NOZZLE_PARK_XY_FEEDRATE 100 // X and Y axes feedrate in mm/s (a
24     #define NOZZLE_PARK_Z_FEEDRATE 5 // Z axis feedrate in mm/s (not used)
25 #endif
26
27 /**
28 * Clean Nozzle Feature — EXPERIMENTAL
29 *
30 * Adds the G12 command to perform a nozzle cleaning process.
31 *
32 * Parameters:

```

```

1   * P  Pattern
2   * S  Strokes / Repetitions
3   * T  Triangles (P1 only)
4   *
5   * Patterns:
6   * P0 Straight line (default). This process requires a sponge type mate
7   *      at a fixed bed location. "S" specifies strokes (i.e. back-forth m
8   *      between the start / end points.
9   *
10  * P1 Zig-zag pattern between (X0, Y0) and (X1, Y1), "T" specifies the
11  *      number of zig-zag triangles to do. "S" defines the number of strok
12  *      Zig-zags are done in whichever is the narrower dimension.
13  *      For example, "G12 P1 S1 T3" will execute:
14  *
15  *      —
16  *      | (X0, Y1) | /\ | /\ | /\ | (X1, Y1)
17  *      |           | / \ | / \ | / \ |
18  *      A |           | / \ | / \ | / \ |
19  *      |           | / \ | / \ | / \ |
20  *      | (X0, Y0) | / \ | / \ | / \ | (X1, Y0)
21  *      — +-----+-----+-----+
22  *                  | ----- | ----- | ----- |
23  *                  T1      T2      T3
24  *
25  * P2 Circular pattern with middle at NOZZLE_CLEAN_CIRCLE_MIDDLE.
26  *      "R" specifies the radius. "S" specifies the stroke count.
27  *      Before starting, the nozzle moves to NOZZLE_CLEAN_START_POINT.
28  *
29  *      Caveats: The ending Z should be the same as starting Z.
30  *      Attention: EXPERIMENTAL. G-code arguments may change.
31  *
32  */

```

```

1 // #define NOZZLE_CLEAN_FEATURE
2
3 #if ENABLED(NOZZLE_CLEAN_FEATURE)
4     // Default number of pattern repetitions
5     #define NOZZLE_CLEAN_STROKES    12
6
7     // Default number of triangles
8     #define NOZZLE_CLEAN_TRIANGLES  3
9
10    // Specify positions as { X, Y, Z }
11    #define NOZZLE_CLEAN_START_POINT { 30, 30, (Z_MIN_POS + 1) }
12    #define NOZZLE_CLEAN_END_POINT   {100, 60, (Z_MIN_POS + 1) }
13
14    // Circular pattern radius
15    #define NOZZLE_CLEAN_CIRCLE_RADIUS 6.5
16    // Circular pattern circle fragments number
17    #define NOZZLE_CLEAN_CIRCLE_FN 10
18    // Middle point of circle
19    #define NOZZLE_CLEAN_CIRCLE_MIDDLE NOZZLE_CLEAN_START_POINT
20
21    // Moves the nozzle to the initial position
22    #define NOZZLE_CLEAN_GOBACK
23 #endif
24
25 /**
26 * Print Job Timer
27 *
28 * Automatically start and stop the print job timer on M104/M109/M190.
29 *
30 *      M104 (hotend, no wait) - high temp = none,           low temp = stop time
31 *      M109 (hotend, wait)    - high temp = start timer, low temp = stop time
32 *      M190 (bed, wait)      - high temp = start timer, low temp = none

```

```

1   *
2   * The timer can also be controlled with the following commands:
3   *
4   *     M75 – Start the print job timer
5   *     M76 – Pause the print job timer
6   *     M77 – Stop the print job timer
7   */
8 #define PRINTJOB_TIMER_AUTOSTART
9
10 /**
11  * Print Counter
12  *
13  * Track statistical data such as:
14  *
15  *     – Total print jobs
16  *     – Total successful print jobs
17  *     – Total failed print jobs
18  *     – Total time printing
19  *
20  * View the current statistics with M78.
21 */
22 #define PRINTCOUNTER
23
24 //=====
25 //===== LCD and SD support =====
26 //=====
27
28 // @section lcd
29
30 /**
31  * LCD LANGUAGE
32 */

```

```

1  * Select the language to display on the LCD. These languages are available
2  *
3  * en , an , bg , ca , cn , cz , cz_utf8 , de , el , el-gr , es , es_utf8 , eu ,
4  * fi , fr , fr_utf8 , gl , hr , it , kana , kana_utf8 , ko_KR , nl , pl , pt ,
5  * pt_utf8 , pt-br , pt-br_utf8 , ru , sk_utf8 , tr , uk , zh_CN , zh_TW , test
6  *
7  * :{ 'en': 'English' , 'an': 'Aragonese' , 'bg': 'Bulgarian' , 'ca': 'Catalan' ,
8  */
9 // #define LCDLANGUAGE en
10
11 /**
12 * LCD Character Set
13 *
14 * Note: This option is NOT applicable to Graphical Displays.
15 *
16 * All character-based LCDs provide ASCII plus one of these
17 * language extensions:
18 *
19 * - JAPANESE ... the most common
20 * - WESTERN ... with more accented characters
21 * - CYRILLIC ... for the Russian language
22 *
23 * To determine the language extension installed on your controller:
24 *
25 * - Compile and upload with LCDLANGUAGE set to 'test'
26 * - Click the controller to view the LCD menu
27 * - The LCD will display Japanese, Western, or Cyrillic text
28 *
29 * See http://marlinfw.org/docs/development/lcd\_language.html
30 *
31 * :[ 'JAPANESE' , 'WESTERN' , 'CYRILLIC' ]
32 */

```

```

1 //#define DISPLAY_CHARSET_HD44780 JAPANESE
2
3 /**
4 * SD CARD
5 *
6 * SD Card support is disabled by default. If your controller has an SD slot
7 * you must uncomment the following option or it won't work.
8 *
9 */
10 #define SDSUPPORT
11
12 /**
13 * SD CARD: SPI SPEED
14 *
15 * Enable one of the following items for a slower SPI transfer speed.
16 * This may be required to resolve "volume init" errors.
17 */
18 //#define SPI_SPEED SPI_HALF_SPEED
19 //#define SPI_SPEED SPI_QUARTER_SPEED
20 //#define SPI_SPEED SPI_EIGHTH_SPEED
21
22 /**
23 * SD CARD: ENABLE CRC
24 *
25 * Use CRC checks and retries on the SD communication.
26 */
27 //#define SD_CHECK_AND_RETRY
28
29 /**
30 * LCD Menu Items
31 *
32 * Disable all menus and only display the Status Screen, or

```

```

1  * just remove some extraneous menu items to recover space .
2  */
3 // #define NO_LCD_MENUS
4 // #define SLIM_LCD_MENUS
5
6 //
7 // ENCODER SETTINGS
8 //
9 // This option overrides the default number of encoder pulses needed to
10 // produce one step. Should be increased for high-resolution encoders.
11 //
12 // #define ENCODER_PULSES_PER_STEP 4
13
14 //
15 // Use this option to override the number of step signals required to
16 // move between next/prev menu items.
17 //
18 // #define ENCODER_STEPS_PER_MENU_ITEM 1
19
20 /**
21 * Encoder Direction Options
22 *
23 * Test your encoder's behavior first with both options disabled .
24 *
25 * Reversed Value Edit and Menu Nav? Enable REVERSE_ENCODER_DIRECTION .
26 * Reversed Menu Navigation only? Enable REVERSE_MENU_DIRECTION .
27 * Reversed Value Editing only? Enable BOTH options .
28 */
29
30 //
31 // This option reverses the encoder direction everywhere .
32 //

```

```

1 // Set this option if CLOCKWISE causes values to DECREASE
2 //
3 // #define REVERSE_ENCODER_DIRECTION
4
5 //
6 // This option reverses the encoder direction for navigating LCD menus.
7 //
8 // If CLOCKWISE normally moves DOWN this makes it go UP.
9 // If CLOCKWISE normally moves UP this makes it go DOWN.
10 //
11 // #define REVERSE_MENU_DIRECTION
12
13 //
14 // Individual Axis Homing
15 //
16 // Add individual axis homing items (Home X, Home Y, and Home Z) to the LCD
17 //
18 // #define INDIVIDUAL_AXIS_HOMING_MENU
19
20 //
21 // SPEAKER/BUZZER
22 //
23 // If you have a speaker that can produce tones , enable it here .
24 // By default Marlin assumes you have a buzzer with a fixed frequency .
25 //
26 #define SPEAKER
27
28 //
29 // STARTUP CHIME
30 //
31 // Play a (non-earpiercing) startup chime on startup/serial connection
32 // of the Trigorilla board

```

```

1  // 
2  // #define STARTUP_CHIME
3
4  //
5  // ENDSTOP BEEP
6  //
7  // Short 2KHz beep when endstops are hit
8  //
9  // #define ENDSTOP_BEEP
10
11 //
12 // The duration and frequency for the UI feedback sound.
13 // Set these to 0 to disable audio feedback in the LCD menus.
14 //
15 // Note: Test audio output with the G-Code:
16 // M300 S<frequency Hz> P<duration ms>
17 //
18 // #define LCD_FEEDBACK_FREQUENCY_DURATION_MS 2
19 // #define LCD_FEEDBACK_FREQUENCY_HZ 5000
20
21 //=====
22 //===== LCD / Controller Selection =====
23 //===== (Character-based LCDs) =====
24 //=====
25
26 //
27 // RepRapDiscount Smart Controller .
28 // http://reprap.org/wiki/RepRapDiscount_Smart_Controller
29 //
30 // Note: Usually sold with a white PCB.
31 //
32 // #define REPRAP_DISCOUNT_SMART_CONTROLLER

```

```

1
2  //
3 // Ultimaker Controller .
4 //
5 // #define ULTIMAKERCONTROLLER
6
7 //
8 // Ultipanel as seen on Thingiverse .
9 //
10 // #define ULTIPANEL
11
12 //
13 // PanelOne from T3P3 ( via RAMPS 1.4 AUX2/AUX3 )
14 // http://reprap.org/wiki/PanelOne
15 //
16 // #define PANEL_ONE
17
18 //
19 // GADGETS3D G3D LCD/SD Controller
20 // http://reprap.org/wiki/RAMPS_1.3/1.4_GADGETS3D_Shield_with_Panel
21 //
22 // Note: Usually sold with a blue PCB.
23 //
24 // #define G3D_PANEL
25
26 //
27 // RigidBot Panel V1.0
28 // http://www.inventapart.com/
29 //
30 // #define RIGIDBOT_PANEL
31
32 //

```

```

1 // Makeboard 3D Printer Parts 3D Printer Mini Display 1602 Mini Controller
2 // https://www.aliexpress.com/item/Micromake-Makeboard-3D-Printer-Parts-3D-
3 //
4 //#define MAKEBOARD_MINI_2_LINE_DISPLAY_1602
5
6 //
7 // ANET and Tronxy 20x4 Controller
8 //
9 //#define ZONESTARLCD           // Requires ADC_KEYPAD_PIN to be assigned
10 // This LCD is known to be susceptible to
11 // which scrambles the display.
12 Pressing any button clears it up.
13 // This is a LCD2004 display with 5 analog
14 //
15 //
16 // Generic 16x2, 16x4, 20x2, or 20x4 character-based LCD.
17 //
18 //#define ULTRALCD
19
20 //=====
21 //===== LCD / Controller Selection =====
22 //===== (I2C and Shift-Register LCDs) =====
23 //=====
24 //
25 //
26 // CONTROLLER TYPE: I2C
27 //
28 // Note: These controllers require the installation of Arduino's LiquidCry
29 // library. For more info: https://github.com/kiyoshigawa/LiquidCrystal_I2C
30 //
31
32 //

```

```

1 // Elefu RA Board Control Panel
2 // http://www.elefu.com/index.php?route=product/product&product_id=53
3 //
4 // #define RA_CONTROL_PANEL
5
6 //
7 // Sainsmart (YwRobot) LCD Displays
8 //
9 // These require F.Malpartida's LiquidCrystal_I2C library
10 // https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/Home
11 //
12 // #define LCD_SAINSMART_I2C_1602
13 // #define LCD_SAINSMART_I2C_2004
14
15 //
16 // Generic LCM1602 LCD adapter
17 //
18 // #define LCM1602
19
20 //
21 // PANELOLU2 LCD with status LEDs,
22 // separate encoder and click inputs.
23 //
24 // Note: This controller requires Arduino's LiquidTWI2 library v1.2.3 or later
25 // For more info: https://github.com/lincomatic/LiquidTWI2
26 //
27 // Note: The PANELOLU2 encoder click input can either be directly connected
28 // to a pin (if BTN_ENC defined to != -1) or read through I2C (when BTN_ENC == -1)
29 //
30 // #define LCD_I2C_PANELOLU2
31
32 //

```

```

1 // Panucatt VIKI LCD with status LEDs,
2 // integrated click & L/R/U/D buttons , separate encoder inputs .
3 //
4 // #define LCD_I2C_VIKI
5
6 //
7 // CONTROLLER TYPE: Shift register panels
8 //
9
10 //
11 // 2 wire Non-latching LCD SR from https://goo.gl/aJJ4sH
12 // LCD configuration: http://reprap.org/wiki/SAV_3D_LCD
13 //
14 // #define SAV_3DLCD
15
16 //=====
17 //===== LCD / Controller Selection =====
18 //===== (Graphical LCDs) =====
19 //=====
20
21 //
22 // CONTROLLER TYPE: Graphical 128x64 (DOGM)
23 //
24 // IMPORTANT: The U8glib library is required for Graphical Display !
25 // https://github.com/olikraus/U8glib_Arduino
26 //
27
28 //
29 // RepRapDiscount FULL GRAPHIC Smart Controller
30 // http://reprap.org/wiki/RepRapDiscount_Full_Graphic_Smart_Controller
31 //
32 // #define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER

```

```

1
2 // 
3 // ReprapWorld Graphical LCD
4 // https://reprapworld.com/?products_details&products_id/1218
5 //
6 // #define REPRAPWORLDGRAPHICALLCD
7
8 //
9 // Activate one of these if you have a Panucatt Devices
10 // Viki 2.0 or mini Viki with Graphic LCD
11 // http://panucatt.com
12 //
13 // #define VIKI2
14 // #define miniVIKI
15
16 //
17 // MakerLab Mini Panel with graphic
18 // controller and SD support – http://reprap.org/wiki/Mini_panel
19 //
20 // #define MINIPANEL
21
22 //
23 // MaKr3d Makr–Panel with graphic controller and SD support .
24 // http://reprap.org/wiki/MaKr3d_MaKrPanel
25 //
26 // #define MAKRPANEL
27
28 //
29 // Adafruit ST7565 Full Graphic Controller .
30 // https://github.com/eboston/Adafruit-ST7565-Full-Graphic-Controller/
31 //
32 // #define ELB_FULL_GRAPHIC_CONTROLLER

```

```

1
2 // 
3 // BQ LCD Smart Controller shipped by
4 // default with the BQ Hephestos 2 and Witbox 2.
5 //
6 // #define BQLCDSMARTCONTROLLER
7
8 //
9 // Cartesio UI
10 // http://mauk.cc/webshop/cartesio-shop/electronics/user-interface
11 //
12 // #define CARTESIO_UI
13
14 //
15 // LCD for Melzi Card with Graphical LCD
16 //
17 // #define LCD_FOR_MELZI
18
19 //
20 // SSD1306 OLED full graphics generic display
21 //
22 // #define U8GLIB_SSD1306
23
24 //
25 // SAV OLEd LCD module support using either SSD1306 or SH1106 based LCD mo
26 //
27 // #define SAV_3DGLCD
28 #if ENABLED(SAV_3DGLCD)
29     // #define U8GLIB_SSD1306
30     #define U8GLIB_SH1106
31 #endif
32

```

```

1  //
2  // Original Ulticontroller from Ultimaker 2 printer with SSD1309 I2C display
3  // https://github.com/Ultimaker/Ultimaker2/tree/master/1249_Ulticontroller-...
4  //
5  //#define ULTI_CONTROLLER
6
7  //
8  // TinyBoy2 128x64 OLED / Encoder Panel
9  //
10 // #define OLED_PANEL_TINYBOY2
11
12 //
13 // MKS MINI12864 with graphic controller and SD support
14 // http://reprap.org/wiki/MKS_MINI_12864
15 //
16 // #define MKS_MINI_12864
17
18 //
19 // Factory display for Creality CR-10
20 // https://www.aliexpress.com/item/Universal-LCD-12864-3D-Printer-Display-S...
21 //
22 // This is RAMPS-compatible using a single 10-pin connector.
23 // (For CR-10 owners who want to replace the Melzi Creality board but retain
24 //
25 // #define CR10_STOCKDISPLAY
26
27 //
28 // ANET and Tronxy Graphical Controller
29 //
30 // #define ANET_FULL_GRAPHICS_LCD // Anet 128x64 full graphics lcd with rotation
31 // // A clone of the RepRapDiscount full graphics LCD
32 // // different pins/wiring (see pins_ANET-12864.h)

```

```

1
2 // 
3 // MKS OLED 1.3" 128 × 64 FULL GRAPHICS CONTROLLER
4 // http://reprap.org/wiki/MKS_12864OLED
5 //
6 // Tiny , but very sharp OLED display
7 //
8 // #define MKS_12864OLED           // Uses the SH1106 controller (default)
9 // #define MKS_12864OLED_SSD1306   // Uses the SSD1306 controller

10
11 //
12 // Silvergate GLCD controller
13 // http://github.com/android444/Silvergate
14 //
15 // #define SILVER_GATE_GLCD_CONTROLLER

16
17 //=====
18 //===== Other Controllers =====
19 //=====

20
21 //
22 // CONTROLLER TYPE: Standalone / Serial
23 //
24
25 //
26 // LCD for Malyan M200 printers .
27 // This requires SDSUPPORT to be enabled
28 //
29 // #define MALYANLCD

30
31 //
32 // CONTROLLER TYPE: Keypad / Add-on

```

```

1  // 
2
3  //
4  // RepRapWorld REPRAPWORLDKEYPAD v1.1
5  // http://reprapworld.com/?products_details&products_id=202&cPath=1591_1626
6  //
7  // REPRAPWORLDKEYPADMOVESTEP sets how much should the robot move when a
8  // is pressed , a value of 10.0 means 10mm per click .
9  //
10 // #define REPRAPWORLDKEYPAD
11 // #define REPRAPWORLDKEYPADMOVESTEP 10.0
12
13 //=====
14 //===== Extra Features =====
15 //=====
16
17 // @section extras
18
19 // Increase the FAN PWM frequency . Removes the PWM noise but increases heat
20 // #define FAST_PWM_FAN
21
22 // Use software PWM to drive the fan , as for the heaters . This uses a very
23 // which is not as annoying as with the hardware PWM . On the other hand , if
24 // is too low , you should also increment SOFT_PWMSCALE .
25 // #define FAN_SOFT_PWM
26
27 // Incrementing this by 1 will double the software PWM frequency ,
28 // affecting heaters , and the fan if FAN_SOFT_PWM is enabled .
29 // However , control resolution will be halved for each increment ;
30 // at zero value , there are 128 effective control positions .
31 #define SOFT_PWMSCALE 0
32

```

```

1 // If SOFT_PWMSCALE is set to a value higher than 0, dithering can
2 // be used to mitigate the associated resolution loss. If enabled,
3 // some of the PWM cycles are stretched so on average the desired
4 // duty cycle is attained.
5 // #define SOFT_PWM_DITHER
6
7 // Temperature status LEDs that display the hotend and bed temperature.
8 // If all hotends, bed temperature, and target temperature are under 54C
9 // then the BLUE led is on. Otherwise the RED led is on. (1C hysteresis)
10 // #define TEMP_STAT_LEDS
11
12 // M240 Triggers a camera by emulating a Canon RC-1 Remote
13 // Data from: http://www.doc-diy.net/photo/rc-1_hacked/
14 // #define PHOTOGRAPH_PIN 23
15
16 // SkeinForge sends the wrong arc g-codes when using Arc Point as fillet pr
17 // #define SF_ARC_FIX
18
19 // Support for the BariCUDA Paste Extruder
20 // #define BARICUDA
21
22 // Support for BlinkM/CyzRgb
23 // #define BLINKM
24
25 // Support for PCA9632 PWM LED driver
26 // #define PCA9632
27
28 /**
29 * RGB LED / LED Strip Control
30 *
31 * Enable support for an RGB LED connected to 5V digital pins, or
32 * an RGB Strip connected to MOSFETs controlled by digital pins.

```

```

1   *
2   * Adds the M150 command to set the LED (or LED strip) color.
3   * If pins are PWM capable (e.g., 4, 5, 6, 11) then a range of
4   * luminance values can be set from 0 to 255.
5   * For Neopixel LED an overall brightness parameter is also available.
6   *
7   * *** CAUTION ***
8   * LED Strips require a MOSFET Chip between PWM lines and LEDs,
9   * as the Arduino cannot handle the current the LEDs will require.
10  * Failure to follow this precaution can destroy your Arduino!
11  * NOTE: A separate 5V power supply is required! The Neopixel LED needs
12  * more current than the Arduino 5V linear regulator can produce.
13  * *** CAUTION ***
14  *
15  * LED Type. Enable only one of the following two options.
16  *
17  */
18 //#define RGB_LED
19 //#define RGBW_LED
20
21 #if ENABLED(RGB_LED) || ENABLED(RGBW_LED)
22     #define RGB_LED_R_PIN 34
23     #define RGB_LED_G_PIN 43
24     #define RGB_LED_B_PIN 35
25     #define RGB_LED_W_PIN -1
26 #endif
27
28 // Support for Adafruit Neopixel LED driver
29 //#define NEOPIXELLED
30 #if ENABLED(NEOPIXELLED)
31     #define NEOPIXEL_TYPE    NEO_GRBW // NEO_GRBW / NEO_GRB – four/three channel
32     #define NEOPIXEL_PIN      4           // LED driving pin on motherboard 4 => I

```

```

1  #define NEOPixel_PIXELS 30          // Number of LEDs in the strip
2  #define NEOPixel_IS_SEQUENTIAL    // Sequential display for temperature ch
3  #define NEOPixel_BRIGHTNESS 127    // Initial brightness (0–255)
4  // #define NEOPixel_STARTUP_TEST // Cycle through colors at startup
5  #endif
6
7  /**
8   * Printer Event LEDs
9   *
10  * During printing , the LEDs will reflect the printer status:
11  *
12  * — Gradually change from blue to violet as the heated bed gets to target
13  * — Gradually change from violet to red as the hotend gets to temperature
14  * — Change to white to illuminate work surface
15  * — Change to green once print has finished
16  * — Turn off after the print has finished and the user has pushed a button
17  */
18 #if ENABLED(BLINKM) || ENABLED(RGB_LED) || ENABLED(RGBW_LED) || ENABLED(PC
19  #define PRINTER_EVENT_LEDS
20 #endif
21
22 /**
23  * R/C SERVO support
24  * Sponsored by TrinityLabs , Reworked by codexmas
25  */
26
27 /**
28  * Number of servos
29  *
30  * For some servo-related options NUMSERVOS will be set automatically .
31  * Set this manually if there are extra servos needing manual control .
32  * Leave undefined or set to 0 to entirely disable the servo subsystem .

```

```

1   */
2 // #define NUMSERVOS 3 // Servo index starts with 0 for M280 command
3
4 // Delay (in milliseconds) before the next move will start , to give the ser
5 // 300ms is a good value but you can try less delay .
6 // If the servo can't reach the requested position , increase it .
7 #define SERVO_DELAY { 300 }

8
9 // Only power servos during movement , otherwise leave off to prevent jitter
10 // #define DEACTIVATESERVOSAFTERMOVE

11
12 /**
13 * Select your version of the Trigorilla (RAMPS1.4) board here .
14 *
15 * 0 = Default Trigorilla
16 * 1 = Newer Trigorilla v1.1 (first seen late 2018)
17 *
18 * The only major difference is a slight change on the servo pin mapping .
19 * This setting only is relevant if you want to use BLtouch or similar
20 * mods to be used via servo pins .
21 * The new version is to be identified by a "TRIGORILLA1.1" lettering
22 * on the upper left of the PCB silkscreen .
23 */
24 #define TRIGORILLA_VERSION 0

25
26 // Enable Anycubic TFT
27 #define ANYCUBIC_TFT_MODEL
28 #define ANYCUBIC_FILAMENT_RUNOUT_SENSOR
29 // #define ANYCUBIC_TFT_DEBUG

30
31 #endif // CONFIGURATION.H

```