

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、3Dプリンター			
<p>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たなプロダクトを作り出し、人類の想像力を最大化させる上で重要なことだと考える。数ある3Dプリンターの素材の中から、私は氷を素材として造形ができる3Dプリンターの開発を行った。</p> <p>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、多くの人々に親しまれて様々な作品が作られている。しかし、氷の作品は作るのには時間がかかり、彫刻の技術や設備が必要となるため、誰でも簡単に氷の造形物と触れ合えるものではない。そのため、通常の3Dプリンターと近い操作方法で使用ができ、常温で造形が可能かつ、リアルタイムで造形が可能な氷をマテリアルとした3Dプリンターの提案をする。</p> <p>3Dプリンターのスライサーソフトとして世界中で使用されるスライサーソフトウェアである Ultimaker Cura で操作が可能な氷をマテリアルとした3Dプリンターの開発を行った。開発した3Dプリンターは、液体窒素で造形用ベッドを冷やすことで常温の環境下でリアルタイムに造形を行う。また、通常の水と粘性を持たせた水ではどちらの方が氷の造形に適しているのか調査を行った。粘性を持たせた水を使用すため、シリンダーとノズルとの距離が遠くなると、抵抗で粘性を持たせた水が通りにくくなるため、シリンダーとノズルが直結しそこから水を絞りだす機構を開発した。</p> <p>実験では、液体窒素を使い常温の環境下でリアルタイムに造形ができるかの調査と純粹な水を使用した場合と水と砂糖を1:3で混ぜた液体でどちらの方が氷の造形に適しているのか調査を行った。今回開発した3Dプリンターの機構で常温の環境下でリアルタイムに造形することができた。氷をマテリアルとした3Dプリンターでの造形を行う際、水と砂糖を混ぜた粘度を上げた水の方が、適しているということが分かった。また、一般の3Dプリンターのスライサーとして広く使われているスライサーソフト Ultimaker Cura を利用して造形できるため、3Dプリンターを扱えるものなら、少ない学習コストで使用することができる。よって、通常の3Dプリンターと近い操作方法で使用ができ、常温で造形が可能かつ、リアルタイムで造形が可能な氷をマテリアルとした3Dプリンターを開発できた。</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	6
8 2.1 関連研究	6
9 2.2 ゲルを用いて印刷する3Dプリンター[1]	6
10 2.3 Suntory-3D on the Rocks[2]	7
11 2.4 A Layered Fabric 3D Printer for Soft Inter active Objects[3]	8
12 2.5 Additive manufacturing of optically trans-parent glass[4]	9
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]	11
14 2.7 フルカラー3Dプリンター—2D印刷から3D印刷へ—[6]	12
15 2.8 3Dプリンタのセラミックスへの適用[7]	14
16 2.9 3D食用ゲルジェットプリンタによる食品創製[8]	15
17 2.10 積彩[9]	17
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]	18
19 2.12 高速・高精細金属3Dプリンターの開発[11]	18
20 2.13 Robot-Assisted Rapid Prototyping for Ice Structures[12]	20
21 2.14 Elsa:氷を素材とした3dプリンターの開発[13]	21
22 2.15 関連研究のまとめ	23
23 第3章 仮説と提案	25
24 3.1 氷をマテリアルとした3Dプリンター	25
25 第4章 機構の実装	27
26 4.1 マテリアルの検討	27
27 4.2 予備実験	28
28 4.3 造形の仕組み	31

14.4	液体窒素を用いた造形の実装	32
24.5	プリンターの本体	32
34.6	プリンターの制御	33
44.7	シリンジの機構	33
54.8	ノズルの機構	34
64.9	ベッドの機構	36
7 第5章	開発したプリンターの精度調査実験	38
85.1	プリンターの動作検証	38
95.2	プリンター動作の確認	38
105.3	氷造形の初期実験	39
115.4	氷の積層実験と結果	39
12 第6章	結果と考察	42
13 第7章	まとめ	44
14	謝辞	47
15	参考文献	48
16 付録A章		50

図 目 次

1

2	1.1 3D プリンターの造形方法一覧	2
3	1.2 印刷速度と精度の関係	3
4	2.1 3D ゲルプリンターで作製した造形物	6
5	2.2 実際に掘削して作製した高精度の彫刻の例	7
6	2.3 実際に印刷された布の造形物	8
7	2.4 ガラス造形の仕組み	9
8	2.5 制作中の様子	10
9	2.6 造形されたガラスの造形物	10
10	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	11
11	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	12
12	2.9 フルカラー 3D プリンターの概略図	13
13	2.10 2D プリンターとフルカラー 3D プリンターの違い	13
14	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	14
15	2.12 SLM 法を用いた印刷で完成した造形物	15
16	2.13 3D 食用ゲルジェットプリンタの模式図	16
17	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	17
18	2.15 積彩によって印刷した造形物	17
19	2.16 インクジェット型金属 3D プリンターの概略図	18
20	2.17 DED の模式図	19
21	2.18 DED を用いて作製した造形物	20
22	2.19 塩水を使ったラフト	21
23	2.20 フロンガスを用いて作製した造形物	22
24	2.21 液体窒素を用いて印刷した造形物	23
25	4.1 砂糖を添加した割合ごとの水	28
26	4.2 設計図と造形した形	29
27	4.3 制作した実際の装置	29
28	4.4 造形過程と完成物	30
29	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（前）	31

1	4.6 開発した氷をマテリアルとしたプリンターの全体図（後）	32
2	4.7 シリンジの機構	35
3	4.8 ノズル	36
4	4.9 液体窒素造形用ベッド	37
5	5.1 押し出し量が多すぎた氷の造形物	40
6	5.2 押し出し量を調整して作られた造形物	41
7	5.3 砂糖を混ぜず水のみでの印刷	41

1

第 1 章

2

はじめに

3 1.1 はじめに

4 3D プリンターは 2010 年代に低価格のものが登場したから今に至るまで注目され続けている。
5 低価格化が進んだことで一般にも普及が進んでいる。3D プリンターを活用することで製造業で
6 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、
7 これまで大量生産により、均一化された商品が流通してきたが、3D プリンターの登場と普及に
8 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
9 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
10 ある。生産者と消費者が同一化により、これまでにない発想の商品が数多く登場し、より便利で
11 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
12 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。しかし、現
13 状の 3D プリンターは印刷できる素材が限られている。新たな 3D プリンターの素材を開発する
14 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、人類の想像力を最大化させるうえ
15 で重要なことだと考える。

16 1.2 現在の 3D プリンター

17 3D プリンターにはいくつかの造形方法がある。主に材料押出法、液槽光重合法、シート積層
18 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。プリ
19 ンターに置ける主な造形方法の機構は図 1.1 に示す。

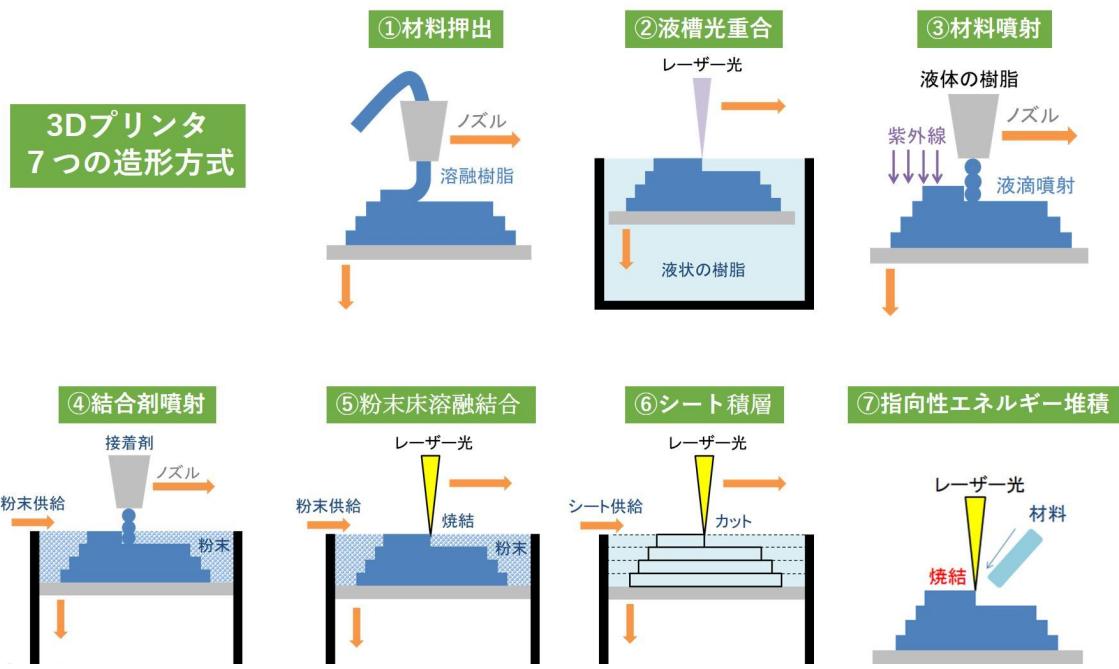


図 1.1 3D プリンターの造形方法一覧

<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、特に熱溶解積層方式 (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、プラスチックをマテリアルとして、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する方法である。一層作るごとに樹脂が固まるまで紫外線を当て続けるため、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 粉末床溶融結合法は、特にレーザー焼結 (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。レーザーで金属粉末に熱を加えるため時間がかかるが、高い精度造形ができる。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

- 1 面を考へ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
2 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。
3 リアルタイムに造形する上で印刷精度と印刷にかかる時間が重要であると考える。それぞれの
4 プリンターがどの程度 2 つの要素を満たしているかを図 1.2 に示す。縦の軸が精度を示し、横の
5 軸が印刷にかかる時間を表す。表を見てわかる通り、ある程度普及している 3D プリンターはあ
6 る程度の造形速度と精度を兼ね備えている。氷の造形物を印刷するプリンターは、速度を犠牲に
7 して正確な造形物を印刷する業務用タイプのみである程度印刷速度を持ったプリンターが無い。
8 そこで、私はリアルタイムで造形が可能な氷をマテリアルとした 3D プリンターを開発した。現
9 在広く普及しているプリンターの一部を改修することで氷造形を可能にする。

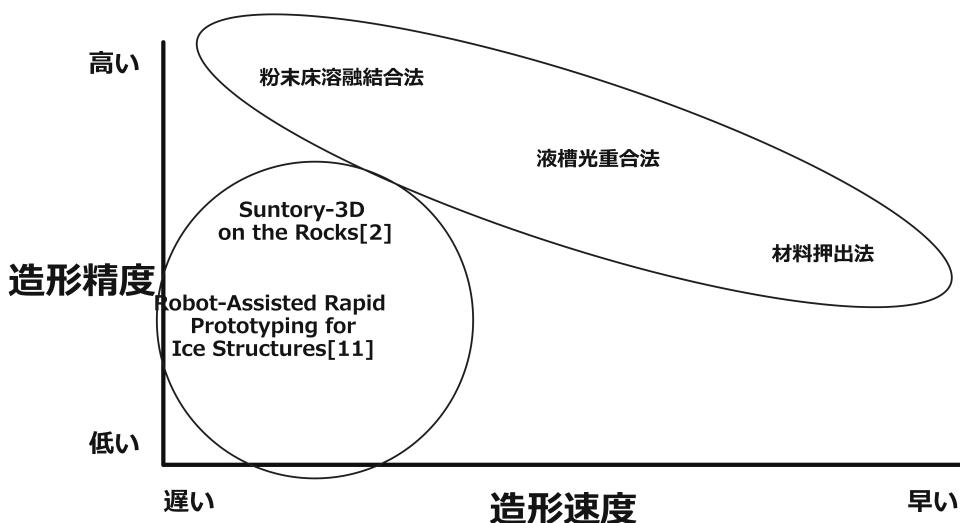


図 1.2 印刷速度と精度の関係

10 1.3 氷をマテリアルとした 3D プリンター

- 11 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
12 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
13 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

1 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
2 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にするため、3D プリン
3 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
4 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
5 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
6 ない。また、Suntory-3D on the Rocks [2] では CNC を使った切削のため特殊な機材と知識が必
7 要になるため、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
8 の多くの人に普及させるためには精度を保つつゝ素早く造形でいるプリンターである必要がある。
9 私は、通常の 3D プリンターと近い操作方法で使用ができる、常温で造形が可能かつ、リアルタイ
10 ムで造形が可能な氷をマテリアルとした 3D プリンターの提案をする。

11 1.4 氷造形の提案

12 氷の造形物をつくる試みは過去に様々な方法で試されてきた。氷をマテリアルとした自動造形
13 には現状 3 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
14 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
15 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
16 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。3 つ目が方に水を流
17 し込み、硬化させる方法である。氷を造形する際に一度型をつくる手間や氷になるまでに数時間
18 と時間が掛かる。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
19 特殊な環境が必要になるため、一般の人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
20 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
21 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
22 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
23 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

1 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
2 られる。また、氷を再定義する。氷をマテリアルとした3Dプリンターの主な使用用途は観賞と
3 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
4 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、氷を下記の2つに
5 定義する。

- 6 1. 固体であるとき常温より低温であること。
7 2. 常温で溶けること。

8 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3Dプリンターと同
9 程度の速度とある程度の精度を両立した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
8 2.1 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
9 とができる。

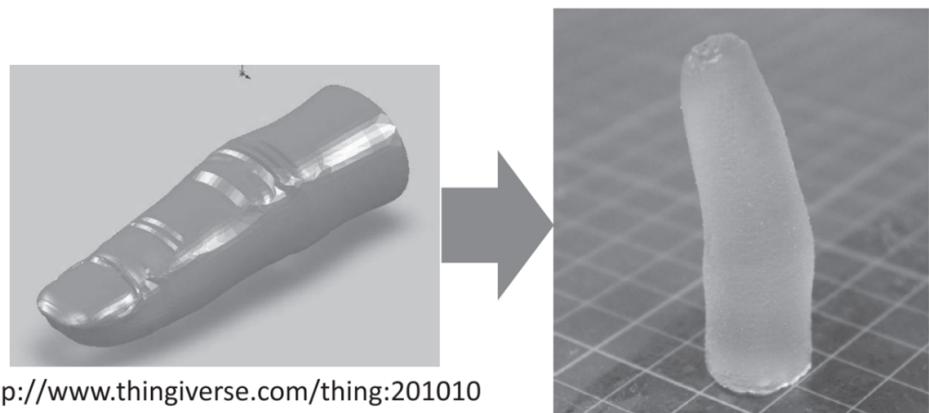


図 2.1 3D ゲルプリンターで作製した造形物

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。制作の過程と実際に制作されたものを図 2.2 に示す。

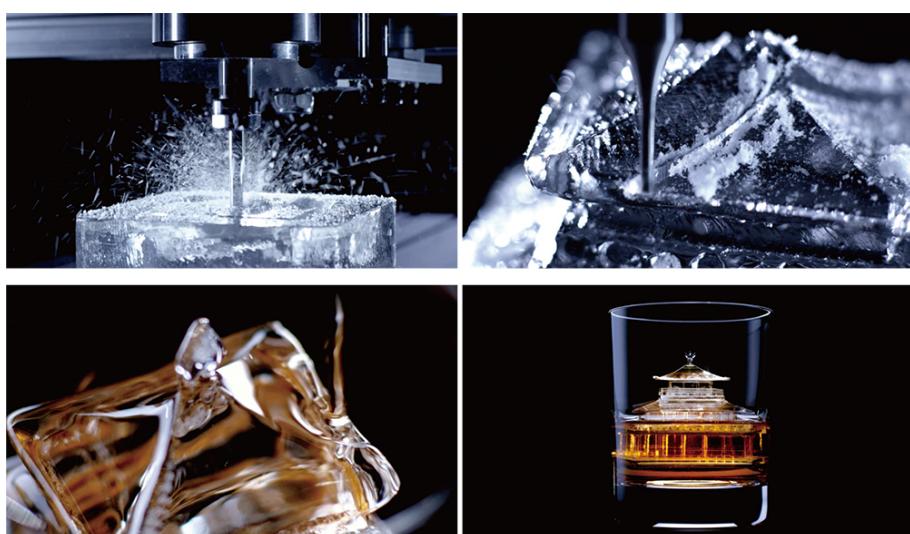


図 2.2 実際に掘削して作製した高精度の彫刻の例

<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。完成した造形物を図 2.3 に示す。

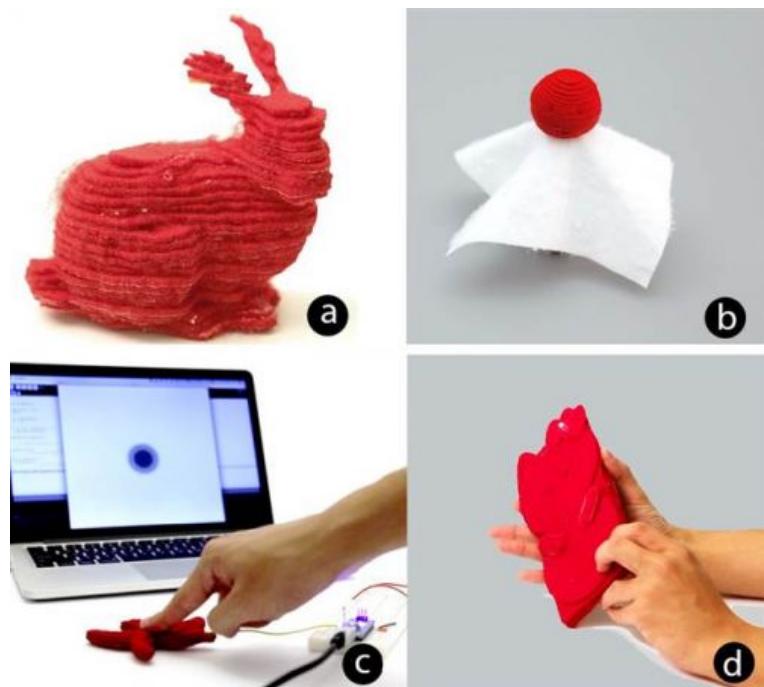


図 2.3 実際に印刷された布の造形物

[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]

制作されてガラスを図 2.6 に提示す。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 1 回の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。造形する仕組みを図 2.4、造形の過程を図 2.6 に示す。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ものになっている。

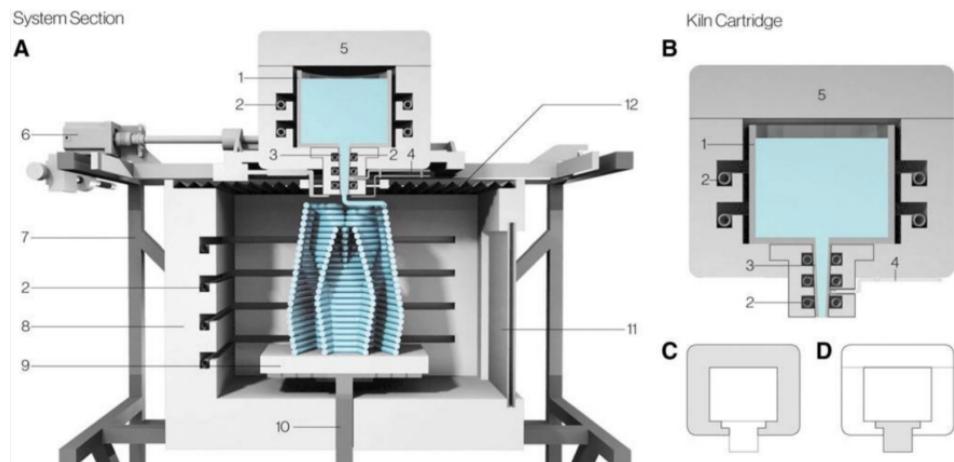


図 2.4 ガラス造形の仕組み



図 2.5 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/)



図 2.6 造形されたガラスの造形物

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

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。下の図 2.7 の様なチョコレートプリンターを作製し、実験を行った。

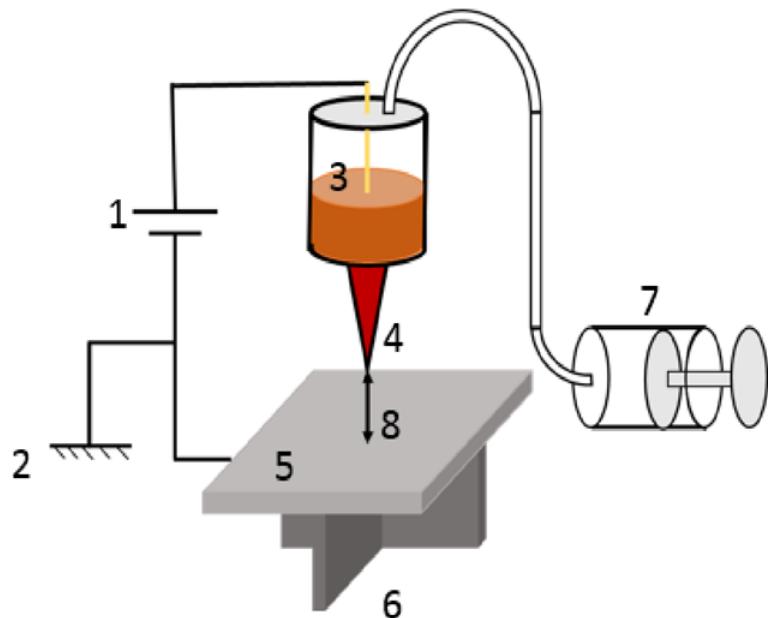


図 2.7 静電インクジェット式プリンターによるチョコレートプリントの仕組み

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出し、下図 2.8 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。

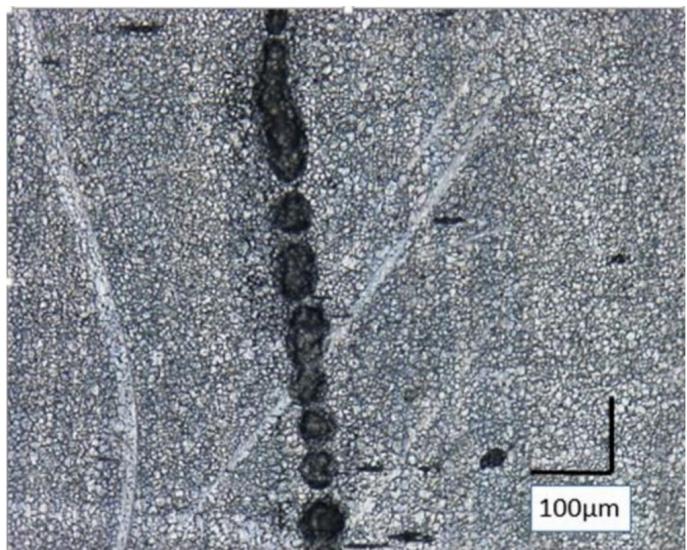


図 2.8 実際に吐出されたチョコレートを顕微鏡を用いて観察した様子

1 2.7 フルカラー 3D プリンター—2D 印刷から 3D 印刷へ—[6]

2 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
3 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
4 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
5 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
6 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。
7 この 3D プリンターは、下図 2.9 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
8 行う。

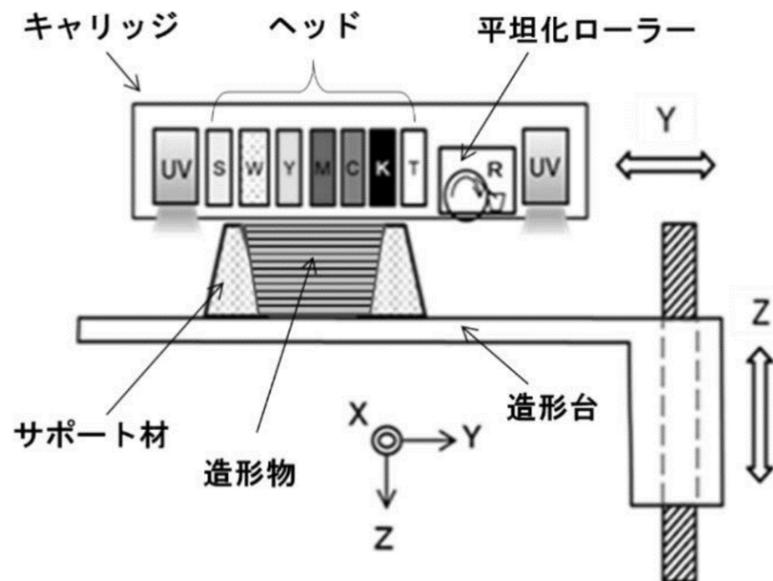


図 2.9 フルカラー 3D プリンターの概略図

1 2D と 3D を比較した時、フルカラー 3D プリンターでは下図 2.10 のような印刷が行われて
2 いる。

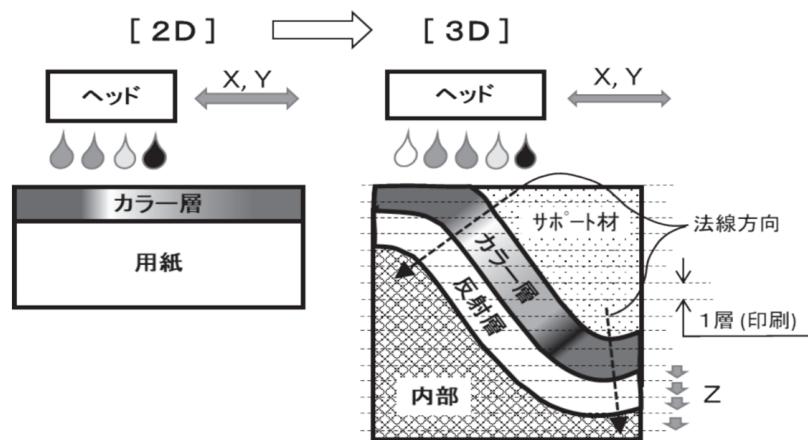


図 2.10 2D プリンターとフルカラー 3D プリンターの違い

3 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
4 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
5 して外形を保つ工夫をしている。

2.8 3D プリンタのセラミックスへの適用 [7]

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。主にセラミックス緻密体を作製する基礎検討に関する研究である。

今回このプリンタには、SLM方式を採用する。この方式では図2.11のように、薄く敷き詰めた粉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。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ているが、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。

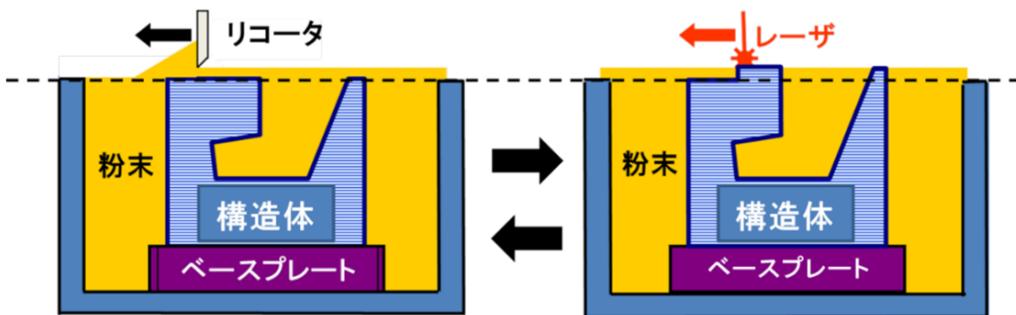


図2.11 SLM法を用いてセラミックスを印刷する際の様子の模式図

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。

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った。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のそれぞれの特性に着目した

①原料セラミックスの選定・粒子径・粒子径分布
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
③造粒粉の作製・流動性・粒子径・かさ密度・1個粒子の密度

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、以下の図2.12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。

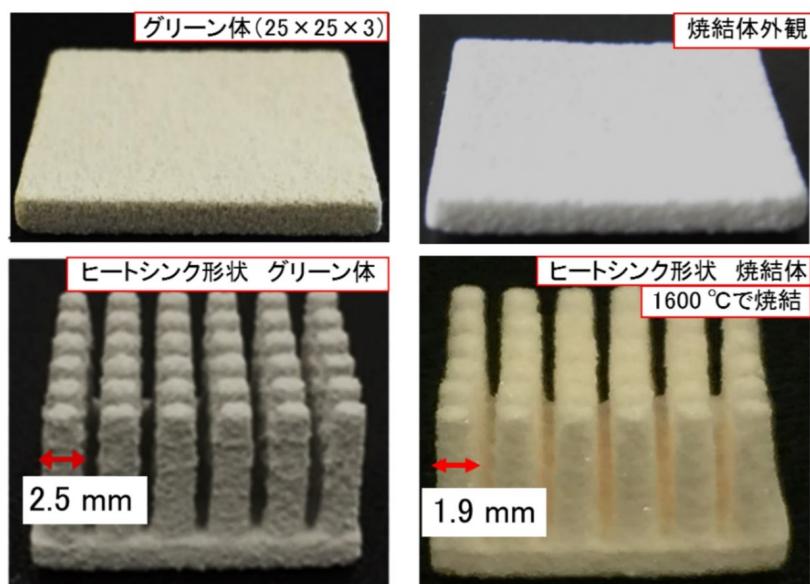


図2.12 SLM法を用いた印刷で完成した造形物

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図2.13のように、①搬送系、②液送系、③冷却系の、三つ

1 の機構から成り立っている。

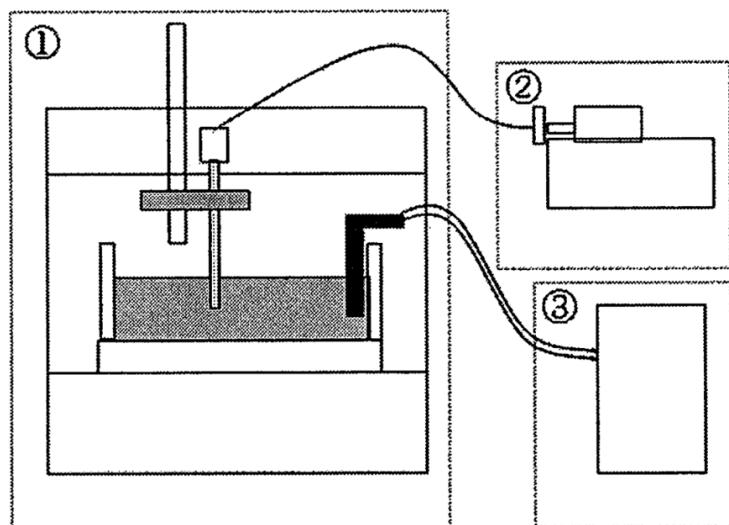


図 2.13 3D 食用ゲルジェットプリンタの模式図

2 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
3 造形することができる機械である。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
4 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
5 ン酸ゲル(人工イクラ)の造形が可能となる。

6 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
7 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
8 が、寒天に比べゼラチンは少し皿にくつついで剥がそうとするとボロボロになった。二つに共通
9 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
10 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。

11 図 2.14 が寒天とゼラチンが固まった様子である。

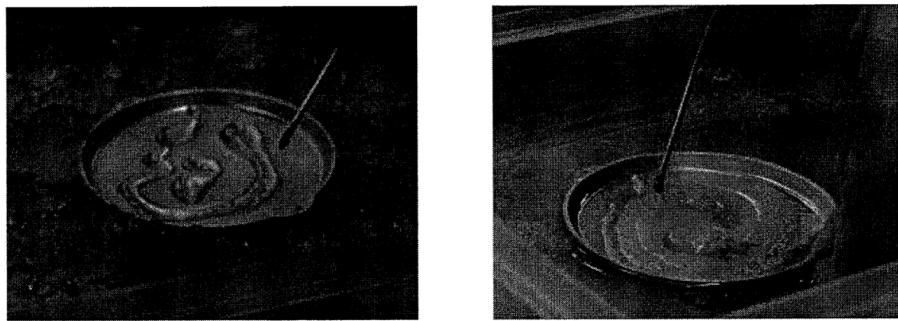


図 2.14 寒天とゼラチンの溶液をそれぞれアルミの皿に垂らしたときの様子

1 2.10 積彩 [9]

2 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 プリンティングの製
3 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」と呼び、また、繊細な織
4 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彩効果）を可能にし、こ
5 の技法を応用して「色瓶」というプロダクト製作している。実際に制作したものが図 2.15 である。



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/>

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]

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 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 プリンターの図 2.16 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。

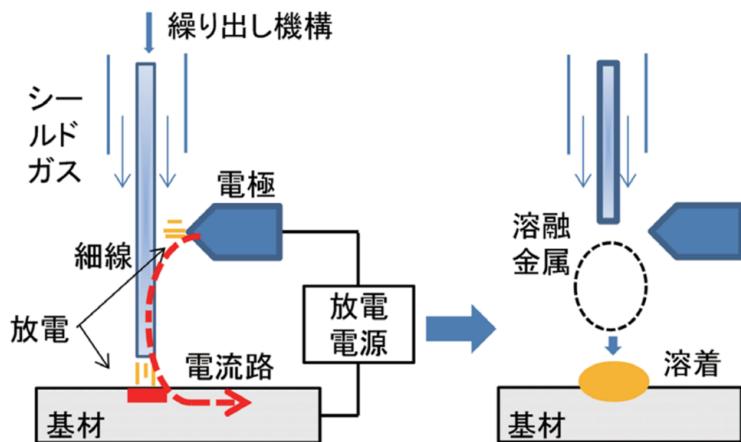


図 2.16 インクジェット型金属 3D プリンターの概略図

2.12 高速・高精細金属 3D プリンターの開発 [11]

この研究では、レーザーメタルでポジションという方法を採用した 3D プリンターの開発を行っている。この方法は DED とも呼ばれていて、レーザー粉末を同時に構造物へ照射し、粉末を溶かしながら造形を進める手法である。以下に印刷している様子の図 2.17 を示す。レーザーを

- 1 ベースプレートに照射すると融解プールが形成される。金属粉末は、ノズルからキャリアガスとともに噴射され、形成された融解収束される。供給された粉末は、プールで溶融し、これを冷却することで凝固する。

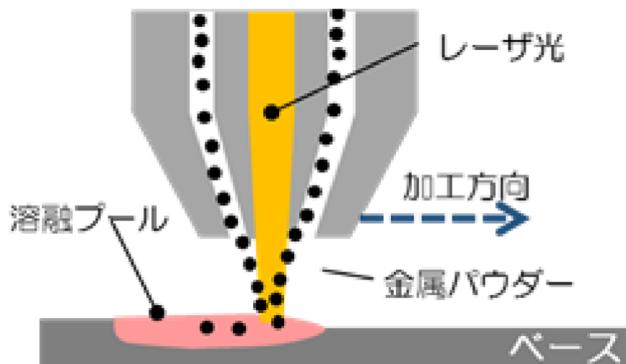


図 2.17 DED の模式図

- 4 従来の金属 3D プリンターの多くは、パウダー・ベッド・フェュージョン (PB) 方式を採用している。この方式では、パウダーベッドを作製し、造形部のみにレーザーを照射し溶融・凝固サイクルとレーザー照射を繰り返して造形を進める。しかし、高速造形のために高出力レーザーを用いると、パウダーのスパッタが多く発生してしまうという課題がある。これにたいし DED は、1. 高出力レーザーが使用できるため造形速度が速い、2. 局所ページを使用することで筐体レス化が可能となり、大型造形にも対応可能、3. 複層造形が可能、などの利点がある。
- 10 また、いかに DED を用いて作製した造形物を図 2.18 に示す。レーザー出力と粉末供給量を増加させることで造形速度が向上し、レーザー出力 4kw のときに、最大で 359cc/h の造形速度を実現した。

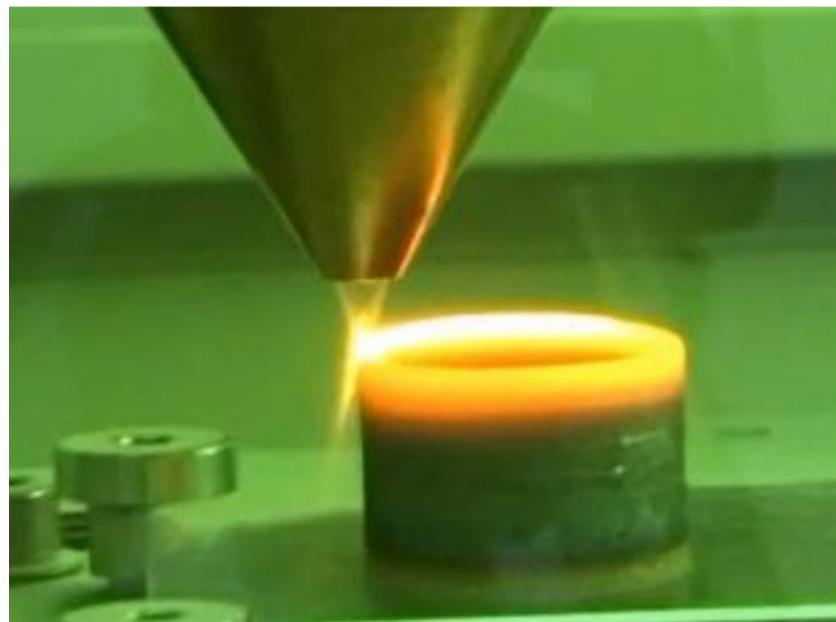


図 2.18 DED を用いて作製した造形物

1 2.13 Robot-Assisted Rapid Prototyping for Ice Structures[12]

2 この研究では、冷やした水を一滴ずつ垂らしながら氷を FDM の様に積み上げて造形する。造
3 形のスピードはかなり遅く 20mm/h で造形するが、精密な造形が可能で塩水をサポート剤として
4 使用し、オーバーハングのある造形も可能になっている。塩水でできた氷は水でできた氷よりも
5 低い温度で溶け始めるため、0 °C の部屋で放置すれば塩水のラフトが溶け氷の造形物だけが残る
6 仕組みになっている。この研究は精度を出すためにスピードを犠牲にしており普通のマグカップ
7 のサイズでも印刷に 50 時間近くかかる。そのため、造形中に溶けないように冷凍庫の中のような
8 環境の部屋で造形する必要がある。実際に造形したものを見ると図 2.19 に示す。



図 2.19 塩水を使ったラフト

1 2.14 Elsa:氷を素材とした 3d プリンターの開発 [13]

2 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
3 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
4 化した代替フロン (HFC134a) を使用する方法を提案している。実際に制作したものを図 2.20 に
5 示す。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
6 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
7 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
8 こと。3 つ目は、環境に対して悪影響があることである。

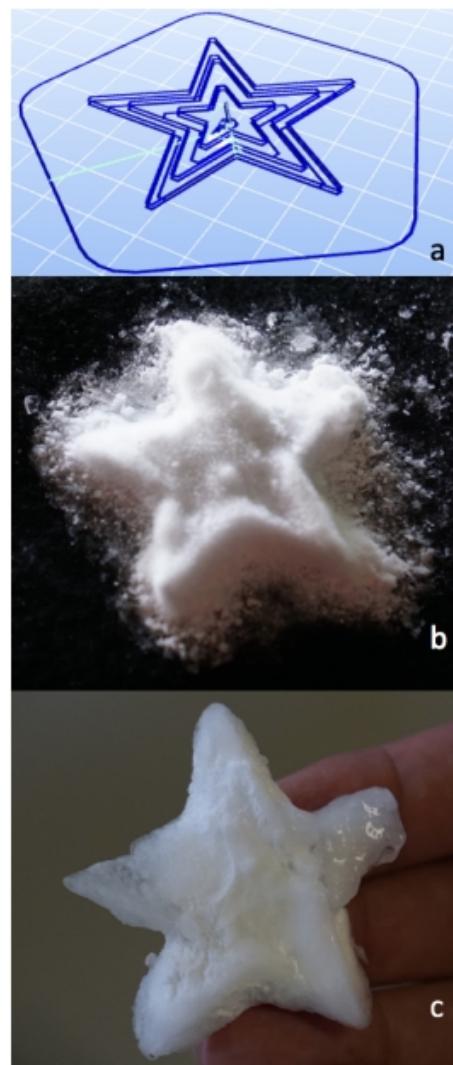


図 2.20 フロンガスを用いて作製した造形物

- 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。液体窒素を
- 3 使い造形したものを図 2.21 に示す。



図 2.21 液体窒素を用いて印刷した造形物

1 2.15 関連研究のまとめ

2 本研究では従来ではない、新たなマテリアルを使った造形をするため、プラスチックではない
3 マテリアルを使い造形物を印刷している研究を多く調べた。関連研究の多くは、常温で安定して
4 いる物質であり、加工が容易なものや温度変化によって性質が変わるもののがマテリアルとして最
5 適なものであることが分かった。また、3D プリンターに関して、近年新たなマテリアルの研究が
6 数多く行われているのと同時に、新しい造形方法の研究が進められていることが分かった。静電
7 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] などでは、静電気を使
8 い、超微細な造形を行おうことが出来る。水も僅かに電荷を帯びているため、この方法を利用
9 して超微細な氷の造形などもできるのではないかと考えた。このように、新しい造形方法が開発
10 されると、これまで造形に適していないと思われていたマテリアルも造形への可能性が生まれるた
11 め、新た造形方法を調査の必要性がある。

12 マテリアルが、プラスチックであるが積彩 [13] の美しさと造形方法は氷をマテリアルとした造
13 形と通じるものがある。美しさでは、氷をマテリアルとした造形では、溶け出することで、一度とし
14 て同じ瞬間が無く、その変化を楽しむという特徴がある。積彩、でも造形物の見る角度を変える
15 と色が変化し、七色のように見える。このように同じ瞬間が無く、変化し続けて見えるもに美し

1 さを感じる部分があるので考える。造形方法についても、複数の色のフィラメントをノズル内で
2 混ぜ、押し出し造形強いる。この技術を発展させれば、Robot-Assisted Rapid Prototyping for
3 Ice Structures [12] のような塩水と水の使い分けが 1 つのノズルできるのではと考える。このよ
4 うな既存の技術の発展についても新たなマテリアルでの造形を行う際の参考になった。

5 本研究は、一般の人でも扱いが可能かつ、一般の 3D プリンターと同程度の速度とある程度の
6 精度を両立した氷をマテリアルとした 3 D プリンターとして、造形精度としては Elsa:氷を素材
7 とした 3d プリンターの開発 [13] と Robot-Assisted Rapid Prototyping for Ice Structures [12]
8 の中間のプリンターを目指し、一般の人でも扱いが可能なプリンターを開発した。

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と水を少しづつたらし長時間かけて、
6 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まうのが問題
7 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。しかし、過冷
8 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まうため、制
9 御が難しい。一般の人でも扱いが可能かつ、一般的な 3D プリンターと同程度の速度とある程度の
10 精度を両立した氷をマテリアルとした 3D プリンターの提案する。必要な要素としては以下のよ
11 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形
7 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り、
8 上記の問題の解決や精度を向上させることができるのでと考える。また、粘度を上げる手段と
9 して、いくつかの方法が考えられる。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
10 ものをマテリアルとして使用する方法だ。シャーベット状のものを使用する場合は、温度管理が
11 必要になるため、今回の機構では、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

3 4.1 マテリアルの検討

4 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
5 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
6 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
7 間に違いをもたらしているのではと考える。

8 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
9 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
10 用した、水と砂糖の割合は 1:3 のものを用意した。水との粘度の比較を図 4.1 に示す。

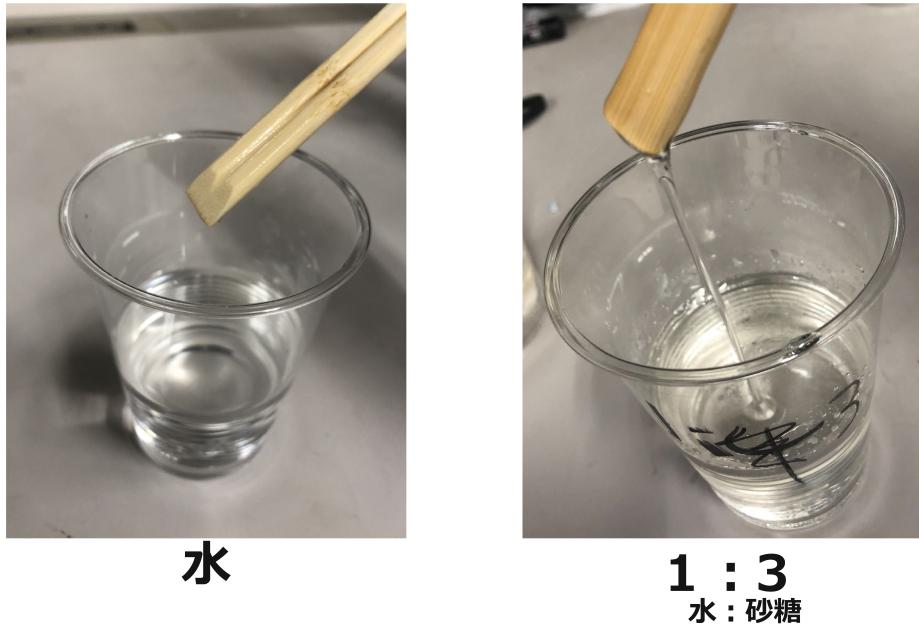


図 4.1 砂糖を添加した割合ごとの水

4.2 予備実験

- 2 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.
- 3 初めに, 3D プリンター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,
- 4 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 4.2 のようになって
- 5 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
- 6 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
- 7 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
- 8 とで, 冷やされた水が氷に変わる. 冷やされた水の温度は 0 度よりも低く, その上に水をたらす
- 9 と氷柱ができるように氷が積層される.

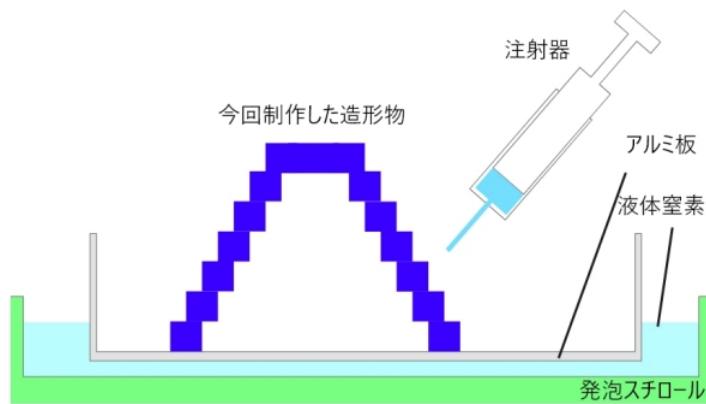


図 4.2 設計図と造形した形



図 4.3 制作した実際の装置

¹ 制作した装置は図 4.3 である。この装置を使い、水の粘度がどの程度が造形速度と造形精度に

1 影響するのか、水を使っての造形と粘度を上げた水で比較を行った。
2 造形物はオーバーハングの調査を行うため図 4.2 のように中を空洞になるように造形を行った。
3 水で造形を行った造形では、高さが 2.0mm 程のしづくがぼつぼつとある状態のものが出来上
4 がり、想定していた形とはほど遠いものになった。また、氷の造形物の上に積層しようと思って
5 も、氷が水をはじいてしまい、積層ができなかった。
6 水の粘度を上げて造形を行ったものが図 4.4 である。造形時間は約 5 分ほどで完成した。造形
7 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た。大きさは横幅約 3 セ
8 ンチ、高さ約 1.5 センチほどである。使用した水の量は、約 100ml ほどである。



図 4.4 造形過程と完成物

9 予備実験では、水の粘度がどの程度が造形速度と造形精度に影響するのか、調査を行った。結
10 果は、関連性があると考えられる。粘度を上げた水の方が、水のみに比べ造形精度が大きく上回っ
11 ていた。また、速度については、水のみの造形では積層ができず比べることができなかったが、粘
12 度のある水では、積層が可能でありオーバーハングも制作できることが確認できた。

4.3 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。また、実際に制作した装置を図 4.5 図 4.6 に示す。

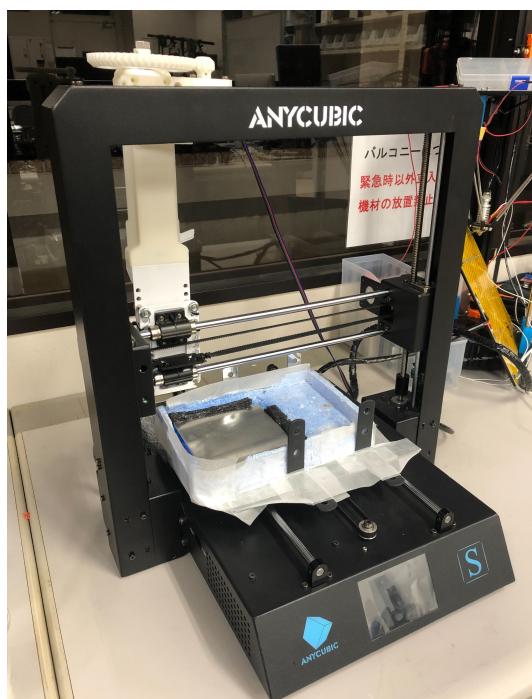


図 4.5 開発した氷をマテリアルとしたプリンターの全体図（前）

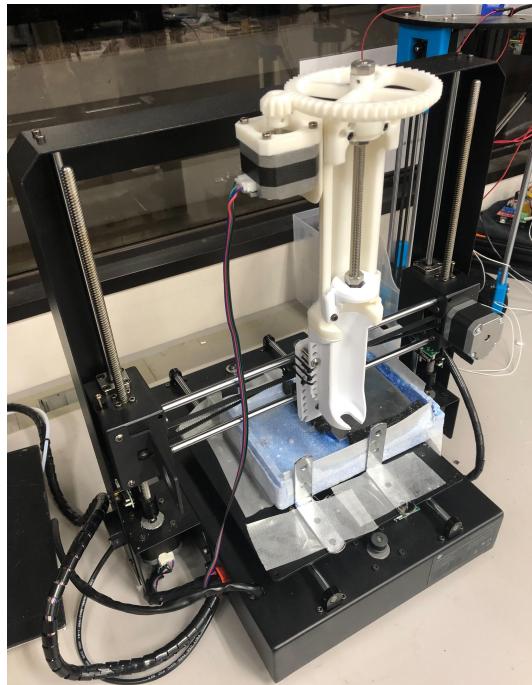


図 4.6 開発した氷をマテリアルとしたプリンターの全体図（後）

4.4 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。

4.5 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。

1 4.6 プリンターの制御

2 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
3 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
4 変更、モータードライバーへの対応、3 D プリンターは安全装置として、一定の温度以下で作動
5 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 3 D プリンターでは必要が無
6 いため、無効にさせた。

7 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
8 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
9 ように制御することができる。

10 4.7 シリンジの機構

11 実際に制作したシリンジを図 4.7 に示す。「60ml Syringe Extruder for Liquid Deposition
12 Modeling - LDM」の機構を一部改造して制作した [14]。シリンジを押し出して水を供給するため
13 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
14 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
15 してしまうと、力不足になることが想定されてため、ギアで回転数を調整している。既存の 3D プ
16 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
17 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。
18 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
19 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。

4.8 ノズルの機構

実際に制作したノズルを図 4.8 に示す。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 である。市販されているシリンダーの先をヒートガンで溶かし、
先を塞いだ後にドリルによって穴を開けた。

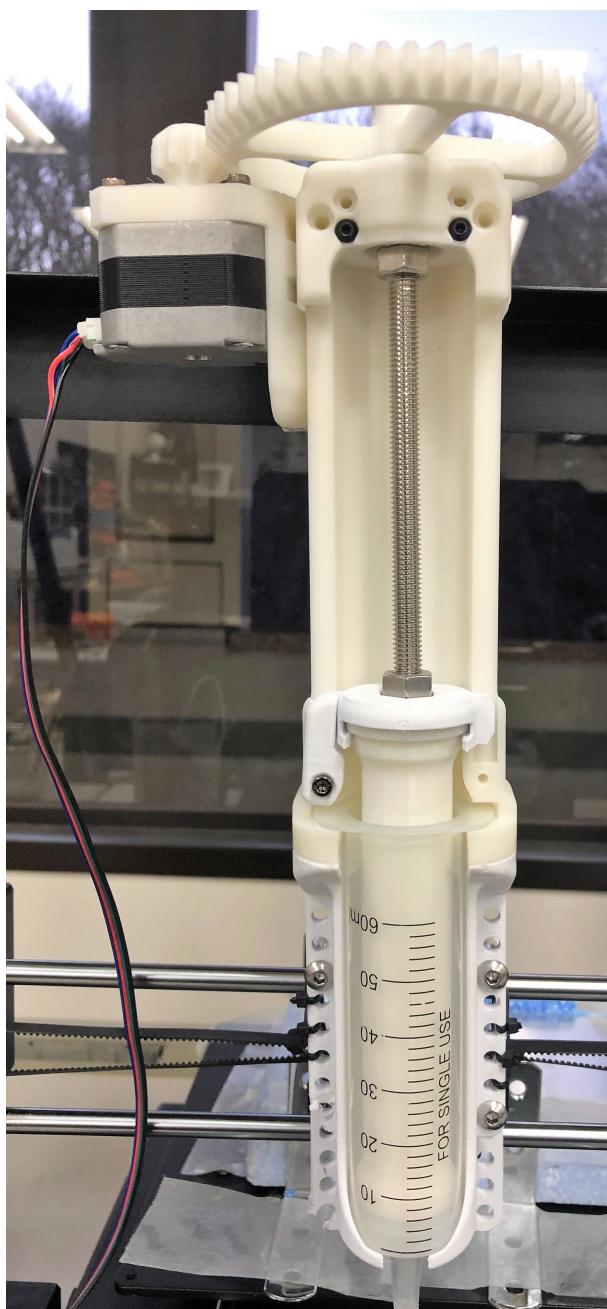


図 4.7 シリンジの機構

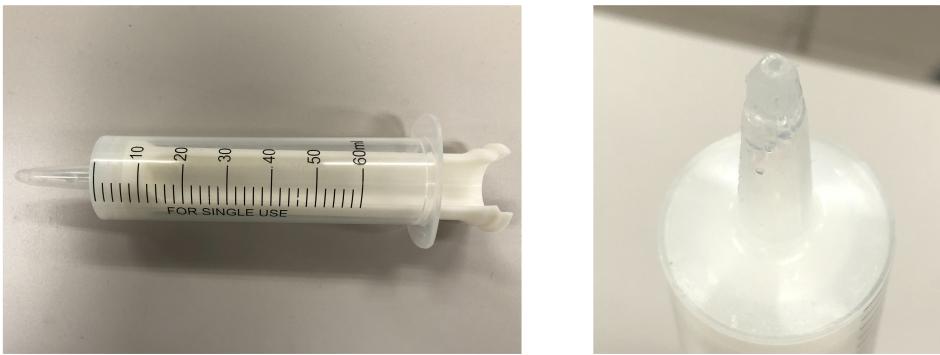


図 4.8 ノズル

1 4.9 ベッドの機構

2 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
3 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
4 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
5 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
6 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
7 を行った。完成したハードは図 4.9 のようになっている。

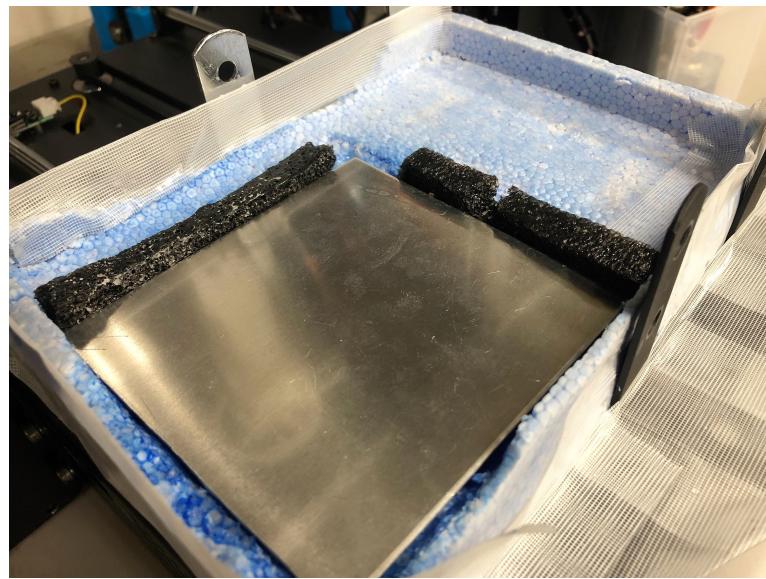


図 4.9 液体窒素造形用ベッド

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になる。

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1:3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、押し出し量:300ということが分かった。また、アルミ
¹² プレートの上に直接水を印刷しようとすると途中でプレートから造形物が剥がれてしまい、印刷
¹³ ができなくなってしまった。その解決として、アルミプレートの上に濡らしたキッチンペーパー¹⁴ を引くことで解決させた。キッチンペーパーを造形物とアルミプレートの間に敷くことで、接着
¹⁵ 面積の増加とキッチンペーパーの纖維に造形物が絡みつくため、外れにくくなつた。また、キッ
¹⁶ チンペーパーも液体窒素により氷ついているため、造形物をキッチンペーパーから剥がす作業も
¹⁷ 容易で、キッチンペーパーは複数回使用することが可能だった。以上の結果とパラメータを基に
¹⁸ 積層の実験を行つた。

¹⁹ 5.4 氷の積層実験と結果

²⁰ 氷造形の初期実験で得られた、パラメーターを基に積層に関する実験を行つた。始めに、ライ
²¹ ン幅:3.0mm、印刷速度:50.0mm、押し出し量:300、レイヤー高さ:1.0mmのパラメータで実
²² 験を行つた。この高さでは、押し出し量に対して、高さが足りず、余分な水が外に飛び出し図5.1

1 のような結果が得られた.

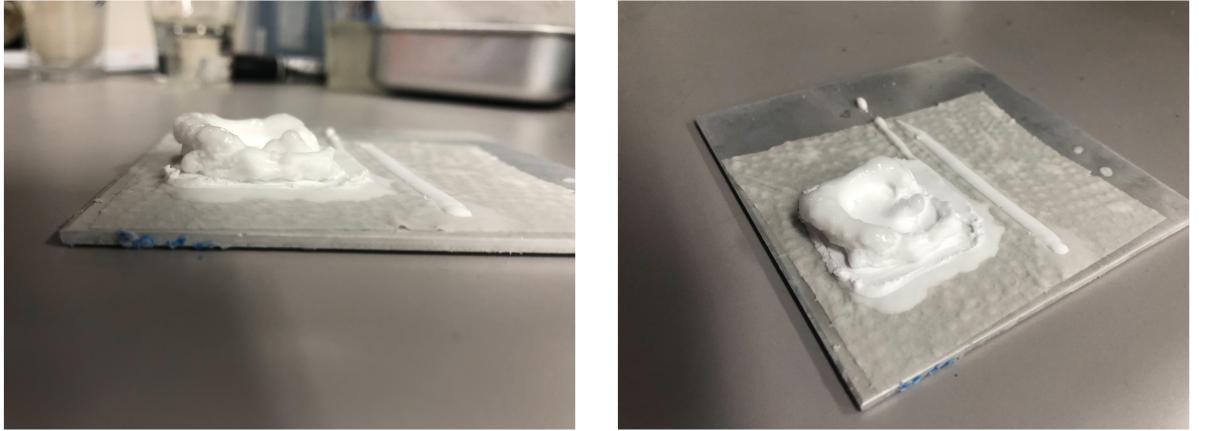


図 5.1 押し出し量が多すぎた氷の造形物

2 上記の失敗を基に、パラメーターの調整を行った。レイヤーの高さを、1.5 mm, 2.0 mm, 2.5 mm,
3 押し出し量のパラメーターを 200, 300, に調整して行ったところ、レイヤー高さを 2.0 mm に変更
4 し、押し出し量を 200 に変更したものが一番綺麗に造形ができるものが図 5.2 である。また、水
5 の押し出し量は、フィラメントの押し出し量を調整することで変更できる、押し出し量の調整は、
6 Arduino IDE を使い、3D プリンターにアクセスできる。Extrusion multiplier のパラメータで
7 押し出し量を調整することができる。造形物の大きさは 20mm × 20mm × 9mm の造形物を印
8 刷できた。



図 5.2 押し出し量を調整して作られた造形物

1 また、水のみで印刷した際の造形が図 5.3 である。水のみの印刷では、水は直ぐに固まってしまう。
2 また、水が一か所に集まろうとして粒が出来上がる。この粒にさらに水が集まり、しづく状
3 の印刷ムラが多くできてしまう。このしづく状の印刷ムラがノズルが造形物に干渉し、積層させ
4 て印刷をすることができなかった。



図 5.3 砂糖を混ぜず水のみでの印刷

第 6 章

結果と考察

今回の実験では、氷をマテリアルとした 3D プリンターの開発と氷のマテリアルとした造形を作成する際、純粋な水ではなく、砂糖を混ぜ粘性を持たせた液体を用いて造形を行った方がより精度の高い造形がでいるのではないかと仮説をたて検証を行った。実験では、純粋な水を使用した場合と水と砂糖を 1 : 3 で混ぜた液体での場合で比較を行った。水と砂糖を混ぜた液体の方が造形精度が高く、積層もスムーズに行われた。よって、氷をマテリアルとした 3D プリンターでの造形を行う際、水と砂糖を混ぜた液体の方が、適しているということが分かった。

また、氷をマテリアルとした 3D プリンターの開発において、使用したものは既存の 3D プリンターの部品やどこでも手に入れやすいものを多く使用した。特に、一般で広く使われているライサーソフト Ultimaker Cura を利用して操作できるようにしたことで、3D プリンターを扱えるの方なら、だれでも操作可能なものになっている。

よって、既存で開発されている氷をマテリアルとした 3D プリンターによりに比べ、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 プリンターの開発ができたと考える。

しかし、今回開発した氷をマテリアルとした 3D プリンターにもいくつか課題が残る。1 つ目が予備実験において、手で試した際の精度を自動かさせた際に再現できていないこと。2 つ目一度に造形できる量に制限があること。この二つを解決できたらと考える。1 つ目の予備実験において、手で試した際の精度を自動かさせた際に再現できていないこととくに、オーバーハングに関しては、いくつか原因が考えられる。手で試した際に使用した、液体の温度が 50 °C 近くと温かかったものに対して、実験で使用したものは常温だった。液体の温度がある程度高温の際、既に造形さ

1 れている氷を一部溶かして積層される。一度解けてから、再度氷になっているため、層ごとの結
2 びつきが強くなる。これと、液体の粘度が加わることにより、予備実験ではオーバーハングが
3 可能であったと考える。そのため、今後押し出す液体の温度についての検証が必要であると考え
4 る。また、今回ノズルの大きさ 1.0mm で行ったが、ほかの大きさでどのような違いが現れるの
5 か、最適なノズルのサイズを探す必要もあると考える。

6 2つ目の一度に造形ができる量が制限があることに関しては、今回液体を押し出す機構とノズル
7 の距離を粘度を持つ流体が通るため、抵抗が大きくなると予想されたため、シリンダーを採用し
8 た。今後ポンプを利用した、液体の押し出し機構などを検討する必要がある。

1

第 7 章

2

まとめ

3 近年 3D プリンターの低価格化が進んだことで一般にも普及が進んでいる。これにより、これ
4 まで生産者と消費者は別の者であったが、生産者と消費者が同一の存在となるなりつつあるのだ。
5 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
6 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。3D プリ
7 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
8 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
9 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
10 だと考える。その中でも、私は氷をマテリアルとした 3D プリンターの開発を行おうと考えた。

11 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、多くの人々に親しまれて様々
12 な作品が作られている。しかし、氷の作品は作るのに時間がかかり、彫刻の技術や設備が必要と
13 なるため、誰でも簡単に触れ合えるものではない。また、現在開発されている氷をマテリアルと
14 した 3D プリンターはマグカップサイズのものを作るのに 50 時間ほどかかるもなど特殊な環境や
15 知識が必要なものしかない。そのため、一般の人でも扱いが可能かつ、一般的の 3D プリンターと同
16 程度の速度とある程度の精度を両立した氷をマテリアルとした 3D プリンターの提案する。

17 一般の人でも扱いが可能かつ、一般的の 3D プリンターと同程度の速度とある程度の精度を両立
18 した氷をマテリアルとした 3D プリンターを開発するにあたり、純粋な水を積層すると、水の粘
19 度が低いため、固まる前に広がってしまう。そのため造形精度が悪く、オーバーハングなども造
20 形することが難しい。よって、水の粘度を上げることにより、上記の問題の解決や精度を向上さ
21 せることができるのではと考えた。また、他の研究では、特殊な機材を使用し、装置が高価にな

りがちである。日本各地で手に入り、価格も 1 リットルあたり 300 円と安価であるため、液体窒素を冷却材として使用する。最後に一般の人でも扱えるように、世界中で使用されている 3D プリントおよびスライ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と考え、氷をマテリアルとした 3D プリンターの開発を行った。開発したプリンターは、液体窒素でベッドを冷やしながら印刷を行う。シリンダーとノズルとの距離が遠くなると、抵抗で粘性を持たせた水が通りにくくなるため、シリンダーとノズルが直結しそこから液体を絞りだす機構を開発した。

実験では、純粋な水を使用した場合と水と砂糖を 1 : 3 で混ぜた液体の場合で比較を行った。水単体と比べ水と砂糖を混ぜた液体の方が造形精度が高く、積層もスムーズに行われた。氷をマテリアルとした 3D プリンターでの造形を行う際、水と砂糖を混ぜた液体の方が、適しているということが分かった。また、開発において使用したものは既存の 3D プリンターの部品やどこでも手に入れやすいものを多く使用した。特に、一般で広く使われているスライサーソフト Ultimaker Cura を利用して造形できるため、3D プリンターを扱えるものなら、少ない学習コストで使用することができる。よって、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 プリンターの開発ができたと考える。

しかし、今回開発した氷をマテリアルとした 3D プリンターにもいくつか課題が残る。1 つ目が予備実験において、手で試した際の精度を自動かさせた際に再現できていないこと。2 つ目一度に造形できる量に制限があること。この二つを解決できたらと考える。

今回の研究開発では、氷をマテリアルとした造形物をある程度の速度と精度で制作を行える機構を開発し、Ultimaker Cura で操作可能なプログラムを組むことで既存の氷をマテリアルとした 3D プリンターに比べ、一般の人でも扱いが可能な 3D プリンターになった水に粘性を持たせることで、一般的の 3D プリンターと同程度の速度とある程度の精度を両立した氷をマテリアルとした 3D プリンターに大きく近づくことができた。しかし、現状のプリンターではまだ、オーバーハングが不十分なこと造形できる量に制限があることなどいくつか問題を抱えている。また、ノズ

- ₁ ルのサイズの調整や水の温度の調整、粘度の調整などまだまだ、ある程度の速度と精度をもった
- ₂ 3D プリンターの開発において、検証可能な要素が残されている。一般の人が気軽に氷の造形を樂
- ₃ しみ、様々な氷の造形物を創造するには、上記の要素の検証を行い、さらなる精度と速度の向上
- ₄ が必要となる。

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. <https://www.codeaward.jp/awards/2014/>
5 work04.html.
- 6 [3] Huaishu Peng, Jennifer Mankoff, Scott E. Hudson, and James McCann. A layered fabric
7 3d printer for soft interactive objects. In *Proceedings of the 33rd Annual ACM Conference*
8 on *Human Factors in Computing Systems*, Vol. 2, pp. 1789–1798, 2015.
- 9 [4] Michael Stern, Giorgia Franchin, Markus Kayser, John Klein, Chikara Inamura, Shreya-
10 Dave, James C Weaver, Peter Houk, Paolo Colombo, Maria Yang, and Neri Oxman.
11 Additive manufacturing of optically transparent glass. *3D Printing and Additive Manu-
12 facturing*, Vol. 2, No. 3, pp. 92–105, 2015.
- 13 [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
14 料の高精度プリント. ライフサポート学会, 2016.
- 15 [6] 八角邦夫. フルカラー 3d プリンター. 科学と教育, 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] 鈴木祐哉, 高岸賢輔, 梅津信二郎. 3d 食用ゲルジェットプリンタによる食品創製. 機械材料・
19 材料加工技術講演会講演論文集, Vol. 20, No. 12-19, 2012.
- 20 [9] 積彩. 積彩. <https://sekisai.com/Grasspiral>.
- 21 [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
22 礎研究. 筑波技術大学テクノレポート, Vol. 25, No. 2, pp. 77–79, 2018.

- 1 [11] 塩見康友, 大野博司, 藤巻晋平, 岡田直忠. 高速・高精細金属 3d プリンタの開発. マイクロエ
2 レクトロニクスシンポジウム論文集, Vol. 25, pp. 271–274, 2018.
- 3 [12] E. Barnett, J. Angeles, D. Pasini, and P.Sijpkes. Robot-assisted rapid proto typing for
4 ice structures. Vol. 25, pp. 146–151, 2009.
- 5 [13] 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
6 タルコンテンツクリエーション (DCC) , Vol. 2017, No. DCC-17, pp. 1–7, 2017.
- 7 [14] piuLAB. 60ml syringe extruder for liquid deposition modeling - ldm. 2014. <https://www.thingiverse.com/thing:482873>.

1

付録 A

2

- 3 本研究のために書き換えた 3D プリンターのファームウェア Marlin 内にある、変更ファイル
4 Configuration.h のコードである。

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

```

1   *
2   */
3
4  /**
5   * Configuration.h
6   *
7   * Basic settings such as:
8   *
9   * - Type of electronics
10  * - Type of temperature sensor
11  * - Printer geometry
12  * - Endstop configuration
13  * - LCD controller
14  * - Extra features
15  *
16  * Advanced settings can be found in Configuration_adv.h
17  *
18  */
19 #ifndef CONFIGURATION_H
20 #define CONFIGURATION_H
21 #define CONFIGURATION_H_VERSION 010109
22
23 //=====
24 //===== Getting Started =====
25 //=====

26
27 /**
28  * Here are some standard links for getting your machine calibrated:
29  *
30  * http://reprap.org/wiki/Calibration
31  * http://youtu.be/wAL9d7FgInk
32  * http://calculator.josefprusa.cz

```

```

1  * http://reprap.org/wiki/Triffid_Hunter%27s_Calibration_Guide
2  * http://www.thingiverse.com/thing:5573
3  * https://sites.google.com/site/repraplogphase/calibration-of-your-reprap
4  * http://www.thingiverse.com/thing:298812
5  */
6
7 //=====
8 //===== DELTA Printer =====
9 //=====
10 // For a Delta printer start with one of the configuration files in the
11 // example_configurations/delta directory and customize for your machine.
12 //
13
14 //=====
15 //===== SCARA Printer =====
16 //=====
17 // For a SCARA printer start with the configuration files in
18 // example_configurations/SCARA and customize for your machine.
19 //
20
21 //=====
22 //===== HANGPRINTER =====
23 //=====
24 // For a Hangprinter start with the configuration file in the
25 // example_configurations/hangprinter directory and customize for your mac
26 //
27
28 // @section info
29
30 // User-specified version info of this build to display in [Pronterface , etc]
31 // startup. Implementation of an idea by Prof Braino to inform user that an
32 // build by the user have been successfully uploaded into firmware.

```

```

1 #define STRING_CONFIG_H_AUTHOR "(davidramiro)" // Who made the changes.
2 #define SHOW_BOOTSCREEN
3 #define STRING_SPLASH_LINE1 SHORT_BUILD_VERSION // will be shown during b
4 #define STRING_SPLASH_LINE2 CUSTOM_BUILD_VERSION // will be shown during b
5
6 /**
7 * *** VENDORS PLEASE READ ***
8 *
9 * Marlin allows you to add a custom boot image for Graphical LCDs.
10 * With this option Marlin will first show your custom screen followed
11 * by the standard Marlin logo with version number and web URL.
12 *
13 * We encourage you to take advantage of this new feature and we also
14 * respectfully request that you retain the unmodified Marlin boot screen.
15 */
16
17 // Enable to show the bitmap in Marlin/_Bootscreen.h on startup .
18 // #define SHOW_CUSTOM_BOOTSCREEN
19
20 // Enable to show the bitmap in Marlin/_Statusscreen.h on the status screen
21 // #define CUSTOM_STATUS_SCREEN_IMAGE
22
23 // @section machine
24
25 /**
26 * Select the serial port on the board to use for communication with the host
27 * This allows the connection of wireless adapters (for instance) to non-USB ports
28 * Serial port 0 is always used by the Arduino bootloader regardless of this setting
29 *
30 * :[0, 1, 2, 3, 4, 5, 6, 7]
31 */
32 #define SERIAL_PORT 0

```

```

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

```

```

1 // :[1, 2, 3, 4, 5]
2 #define EXTRUDERS 1
3
4 // Generally expected filament diameter (1.75, 2.85, 3.0, ...). Used for V
5 #define DEFAULT_NOMINAL_FILAMENT_DIA 1.75
6
7 // For Cyclops or any "multi-extruder" that shares a single nozzle.
8 // #define SINGLENOZZLE
9
10 /**
11 * Prusa MK2 Single Nozzle Multi-Material Multiplexer, and variants.
12 *
13 * This device allows one stepper driver on a control board to drive
14 * two to eight stepper motors, one at a time, in a manner suitable
15 * for extruders.
16 *
17 * This option only allows the multiplexer to switch on tool-change.
18 * Additional options to configure custom E moves are pending.
19 */
20 // #define MK2_MULTIPLEXER
21 #if ENABLED(MK2_MULTIPLEXER)
22     // Override the default DIO selector pins here, if needed.
23     // Some pins files may provide defaults for these pins.
24     // #define E_MUX0_PIN 40 // Always Required
25     // #define E_MUX1_PIN 42 // Needed for 3 to 8 steppers
26     // #define E_MUX2_PIN 44 // Needed for 5 to 8 steppers
27 #endif
28
29 // A dual extruder that uses a single stepper motor
30 // #define SWITCHING_EXTRUDER
31 #if ENABLED(SWITCHING_EXTRUDER)
32     #define SWITCHING_EXTRUDER_SERVO_NR 0

```

```

1 #define SWITCHING_EXTRUDER_SERVO_ANGLES { 0, 90 } // Angles for E0, E1[, ,
2 #if EXTRUDERS > 3
3     #define SWITCHING_EXTRUDER_E23_SERVO_NR 1
4 #endif
5 #endif
6
7 // A dual-nozzle that uses a servomotor to raise/lower one of the nozzles
8 // #define SWITCHING_NOZZLE
9 #if ENABLED(SWITCHING_NOZZLE)
10 #define SWITCHING_NOZZLE_SERVO_NR 0
11 #define SWITCHING_NOZZLE_SERVO_ANGLES { 0, 90 } // Angles for E0, E1
12 // #define HOTEND_OFFSET_Z { 0.0, 0.0 }
13 #endif
14
15 /**
16 * Two separate X-carriages with extruders that connect to a moving part
17 * via a magnetic docking mechanism. Requires SOL1_PIN and SOL2_PIN.
18 */
19 // #define PARKING_EXTRUDER
20 #if ENABLED(PARKING_EXTRUDER)
21 #define PARKING_EXTRUDER_SOLENOIDS_INVERT // If enabled, the so
22 #define PARKING_EXTRUDER_SOLENOIDS_PINS_ACTIVE LOW // LOW or HIGH pin si
23 #define PARKING_EXTRUDER_SOLENOIDS_DELAY 250 // Delay (ms) for ma
24 #define PARKING_EXTRUDER_PARKING_X { -78, 184 } // X positions for pa
25 #define PARKING_EXTRUDER_GRAB_DISTANCE 1 // mm to move beyond
26 #define PARKING_EXTRUDER_SECURITY_RAISE 5 // Z-raise before pa
27 #define HOTEND_OFFSET_Z { 0.0, 1.3 } // Z-offsets of the t
28 #endif
29
30 /**
31 * "Mixing Extruder"
32 * — Adds G-codes M163 and M164 to set and "commit" the current mix facto

```

```

1   *      — Extends the stepping routines to move multiple steppers in proportion
2   *      — Optional support for Repetier Firmware's 'M164 S<index>' supporting
3   *      — This implementation supports up to two mixing extruders.
4   *      — Enable DIRECT_MIXING_IN_G1 for M165 and mixing in G1 (from Pia Taub
5   */
6 // #define MIXING_EXTRUDER
7 #if ENABLED(MIXING_EXTRUDER)
8   #define MIXING_STEPPERS 2           // Number of steppers in your mixing ext
9   #define MIXING_VIRTUAL_TOOLS 16    // Use the Virtual Tool method with M163
10  // #define DIRECT_MIXING_IN_G1    // Allow ABCDHII mix factors in G1 movem
11 #endif
12
13 // Offset of the extruders (uncomment if using more than one and relying on
14 // The offset has to be X=0, Y=0 for the extruder 0 hotend (default extrude
15 // For the other hotends it is their distance from the extruder 0 hotend.
16 // #define HOTEND_OFFSET_X {0.0, 20.00} // (in mm) for each extruder , offset
17 // #define HOTEND_OFFSET_Y {0.0, 5.00} // (in mm) for each extruder , offset
18
19 // @section machine
20
21 /**
22  * Select your power supply here. Use 0 if you haven't connected the PS_ON_
23  *
24  * 0 = No Power Switch
25  * 1 = ATX
26  * 2 = X-Box 360 203Watts (the blue wire connected to PS_ON and the red wire
27  *
28  * :{ 0:'No power switch' , 1:'ATX' , 2:'X-Box 360' }
29  */
30 #define POWER_SUPPLY 0
31
32 #if POWER_SUPPLY > 0

```

```

1 // Enable this option to leave the PSU off at startup.
2 // Power to steppers and heaters will need to be turned on with M80.
3 // #define PS_DEFAULT_OFF
4
5 // #define AUTO_POWER_CONTROL           // Enable automatic control of the PSU
6 #if ENABLED(AUTO_POWER_CONTROL)
7     #define AUTO_POWER_FANS           // Turn on PSU if fans need power
8     #define AUTO_POWER_E_FANS
9     #define AUTO_POWER_CONTROLLERFAN
10    #define POWER_TIMEOUT 30
11 #endif
12
13#endif
14
15// @section temperature
16
17//=====
18//===== Thermal Settings =====
19//=====
20
21 /**
22 * —NORMAL IS 4.7kohm PULLUP!— 1kohm pullup can be used on hotend sensor,
23 *
24 * Temperature sensors available:
25 *
26 * -4 : thermocouple with AD8495
27 * -3 : thermocouple with MAX31855 (only for sensor 0)
28 * -2 : thermocouple with MAX6675 (only for sensor 0)
29 * -1 : thermocouple with AD595
30 * 0 : not used
31 * 1 : 100k thermistor – best choice for EPCOS 100k (4.7k pullup)
32 * 2 : 200k thermistor – ATC Semitec 204GT-2 (4.7k pullup)

```

1 * 3 : Mendel-parts thermistor (4.7k pullup)
2 * 4 : 10k thermistor !! do not use it for a hotend. It gives bad resol
3 * 5 : 100K thermistor – ATC Semitec 104GT-2/104NT-4-R025H42G (Used in
4 * 501 : 100K Zonestar (Tronxy X3A) Thermistor
5 * 6 : 100k EPCOS – Not as accurate as table 1 (created using a fluke t
6 * 7 : 100k Honeywell thermistor 135-104LAG-J01 (4.7k pullup)
7 * 71 : 100k Honeywell thermistor 135-104LAF-J01 (4.7k pullup)
8 * 8 : 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup)
9 * 9 : 100k GE Sensing AL03006-58.2K-97-G1 (4.7k pullup)
10 * 10 : 100k RS thermistor 198-961 (4.7k pullup)
11 * 11 : 100k beta 3950 1% thermistor (4.7k pullup)
12 * 12 : 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup) (calibrated
13 * 13 : 100k Hisens 3950 1% up to ° 300C for hotend "Simple ONE" & "Ho
14 * 15 : 100k thermistor calibration for JGAurora A5 hotend
15 * 20 : the PT100 circuit found in the Ultimainboard V2.x
16 * 60 : 100k Maker's Tool Works Kapton Bed Thermistor beta=3950
17 * 66 : 4.7M High Temperature thermistor from Dyze Design
18 * 70 : the 100K thermistor found in the bq Hephestos 2
19 * 75 : 100k Generic Silicon Heat Pad with NTC 100K MGB18-104F39050L32 t
20 *
21 * 1k ohm pullup tables – This is atypical, and requires changing ou
22 * (but gives greater accuracy and more stable
23 * 51 : 100k thermistor – EPCOS (1k pullup)
24 * 52 : 200k thermistor – ATC Semitec 204GT-2 (1k pullup)
25 * 55 : 100k thermistor – ATC Semitec 104GT-2 (Used in ParCan & J-Head)
26 *
27 * 1047 : Pt1000 with 4k7 pullup
28 * 1010 : Pt1000 with 1k pullup (non standard)
29 * 147 : Pt100 with 4k7 pullup
30 * 110 : Pt100 with 1k pullup (non standard)
31 *
32 * Use these for Testing or Development purposes. NEVER for product

```

1   *      998 : Dummy Table that ALWAYS reads ° 25C or the temperature defined by
2   *      999 : Dummy Table that ALWAYS reads ° 100C or the temperature defined by
3   *
4   * :{ '0': "Not used", '1':"100k / 4.7k - EPCOS", '2':"200k / 4.7k - ATC S
5 1% up to ° 300C for hotend 'Simple ONE' & hotend 'All In ONE', '20':"PT100
6 */
7 #define TEMP_SENSOR_0 5
8 #define TEMP_SENSOR_1 0
9 #define TEMP_SENSOR_2 0
10 #define TEMP_SENSOR_3 0
11 #define TEMP_SENSOR_4 0
12 #define TEMP_SENSOR_BED 1
13 #define TEMP_SENSOR_CHAMBER 0
14
15 // Dummy thermistor constant temperature readings , for use with 998 and 999
16 #define DUMMY_THERMISTOR_998_VALUE 25
17 #define DUMMY_THERMISTOR_999_VALUE 100
18
19 // Use temp sensor 1 as a redundant sensor with sensor 0. If the readings
20 // from the two sensors differ too much the print will be aborted.
21 // #define TEMP_SENSOR_1_AS_REDUNDANT
22 #define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10
23
24 // Extruder temperature must be close to target for this long before M109 returns
25 #define TEMP_RESIDENCY_TIME 10 // (seconds)
26 #define TEMP_HYSTERESIS 3 // (degC) range of +/- temperatures considered
27 #define TEMP_WINDOW 1 // (degC) Window around target to start the
28
29 // Bed temperature must be close to target for this long before M190 returns
30 #define TEMP_BED_RESIDENCY_TIME 10 // (seconds)
31 #define TEMP_BED_HYSTERESIS 3 // (degC) range of +/- temperatures considered
32 #define TEMP_BED_WINDOW 1 // (degC) Window around target to start the

```

```

1
2 // The minimal temperature defines the temperature below which the heater v
3 // to check that the wiring to the thermistor is not broken.
4 // Otherwise this would lead to the heater being powered on all the time.
5 #define HEATER_0_MINTEMP 5
6 #define HEATER_1_MINTEMP 5
7 #define HEATER_2_MINTEMP 5
8 #define HEATER_3_MINTEMP 5
9 #define HEATER_4_MINTEMP 5
10 #define BED_MINTEMP 5
11
12 // When temperature exceeds max temp, your heater will be switched off.
13 // This feature exists to protect your hotend from overheating accidentally.
14 // You should use MINTEMP for thermistor short/failure protection.
15 #define HEATER_0_MAXTEMP 285
16 #define HEATER_1_MAXTEMP 275
17 #define HEATER_2_MAXTEMP 275
18 #define HEATER_3_MAXTEMP 275
19 #define HEATER_4_MAXTEMP 275
20 #define BED_MAXTEMP 135
21
22 //=====
23 //===== PID Settings =====
24 //=====
25 // PID Tuning Guide here: http://reprap.org/wiki/PID\_Tuning
26
27 // Comment the following line to disable PID and enable bang-bang.
28 #define PIDTEMP
29 #define BANGMAX 255      // Limits current to nozzle while in bang-bang mode
30 #define PID_MAX_BANGMAX // Limits current to nozzle while PID is active (so
31 #define PID_K1 0.95       // Smoothing factor within any PID loop
32 #if ENABLED(PIDTEMP)

```

```

1 // #define PID_AUTOTUNE_MENU // Add PID Autotune to the LCD "Temperature"
2 // #define PID_DEBUG // Sends debug data to the serial port.
3 // #define PID_OPENLOOP 1 // Puts PID in open loop. M104/M140 sets the output
4 // #define SLOW_PWM_HEATERS // PWM with very low frequency (roughly 0.125Hz)
5 // #define PID_PARAMS_PER_HOTEND // Uses separate PID parameters for each
6 // // Set/get with gcode: M301 E[extruder number]
7 #define PID_FUNCTIONAL_RANGE 10 // If the temperature difference between
8 // // is more than PID_FUNCTIONAL_RANGE then
9
10 // If you are using a pre-configured hotend then you can use one of the
11
12 // i3 Mega stock v5 hotend, 40W heater cartridge Ω(3.6 @ ° 22C)
13 #define DEFAULT_Kp 15.94
14 #define DEFAULT_Ki 1.17
15 #define DEFAULT_Kd 54.19
16
17 // Ultimaker
18 // #define DEFAULT_Kp 22.2
19 // #define DEFAULT_Ki 1.08
20 // #define DEFAULT_Kd 114
21
22 // MakerGear
23 // #define DEFAULT_Kp 7.0
24 // #define DEFAULT_Ki 0.1
25 // #define DEFAULT_Kd 12
26
27 // Mendel Parts V9 on 12V
28 // #define DEFAULT_Kp 63.0
29 // #define DEFAULT_Ki 2.25
30 // #define DEFAULT_Kd 440
31
32 #endif // PIDTEMP

```

```

1
2 //=====
3 //===== PID > Bed Temperature Control =====
4 //=====

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

```

```

1  #if ENABLED(PIDTEMPBED)
2
3  // #define PID_BED_DEBUG // Sends debug data to the serial port.
4
5  // Anycubic i3 Mega Ultrabase Ω(0.9 @ ° 22C)
6  #define DEFAULT_bedKp 251.78
7  #define DEFAULT_bedKi 49.57
8  #define DEFAULT_bedKd 319.73
9
10 // 120V 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)
11 // from pidautotune
12 // #define DEFAULT_bedKp 97.1
13 // #define DEFAULT_bedKi 1.41
14 // #define DEFAULT_bedKd 1675.16
15
16 // FIND YOUR OWN: "M303 E-1 C8 S90" to run autotune on the bed at 90 deg
17 #endif // PIDTEMPBED
18
19 // @section extruder
20
21 /**
22 * Prevent extrusion if the temperature is below EXTRUDE_MINTEMP.
23 * Add M302 to set the minimum extrusion temperature and/or turn
24 * cold extrusion prevention on and off.
25 *
26 * *** IT IS HIGHLY RECOMMENDED TO LEAVE THIS OPTION ENABLED! ***
27 */
28 #define PREVENT_COLD_EXTRUSION 変えた
29 //
30 #define EXTRUDE_MINTEMP -5
31
32 /**

```

```

1   * Prevent a single extrusion longer than EXTRUDEMAXLENGTH.
2   * Note: For Bowden Extruders make this large enough to allow load/unload.
3   */
4 #define PREVENT_LENGTHY_EXTRUDE
5 #define EXTRUDEMAXLENGTH 600
6
7 //=====
8 //===== Thermal Runaway Protection =====
9 //=====

10 /**
11  *
12  * Thermal Protection provides additional protection to your printer from
13  * and fire. Marlin always includes safe min and max temperature ranges which
14  * protect against a broken or disconnected thermistor wire.
15  *
16  * The issue: If a thermistor falls out, it will report the much lower
17  * temperature of the air in the room, and the firmware will keep
18  * the heater on.
19  *
20  * If you get "Thermal Runaway" or "Heating failed" errors the
21  * details can be tuned in Configuration_adv.h
22  */
23
24 #define THERMALPROTECTIONHOTENDS // Enable thermal protection for all extruders
25 #define THERMALPROTECTIONBED // Enable thermal protection for the build plate
26
27 //=====
28 //===== Mechanical Settings =====
29 //=====

30
31 // @section machine
32

```

```

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

```

```

1 // #define ENDSTOPPULLUP_XMIN
2 // #define ENDSTOPPULLUP_YMIN
3 // #define ENDSTOPPULLUP_ZMIN
4 // #define ENDSTOPPULLUP_ZMIN_PROBE
5 #endif
6
7 // Mechanical endstop with COM to ground and NC to Signal uses "false" here
8 #define X_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of
9 #define Y_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of
10 #define Z_MIN_ENDSTOP_INVERTING true // set to true to invert the logic of
11 #define X_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of
12 #define Y_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of
13 #define Z_MAX_ENDSTOP_INVERTING true // set to true to invert the logic of
14 #define Z_MIN_PROBE_ENDSTOP_INVERTING true // set to true to invert the log
15
16 /**
17 * Stepper Drivers
18 *
19 * These settings allow Marlin to tune stepper driver timing and enable adv
20 * stepper drivers that support them. You may also override timing options
21 *
22 * A4988 is assumed for unspecified drivers.
23 *
24 * Options: A4988, DRV8825, LV8729, L6470, TB6560, TB6600, TMC2100,
25 *          TMC2130, TMC2130_STANDALONE, TMC2208, TMC2208_STANDALONE,
26 *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
27 *          TMC5130, TMC5130_STANDALONE
28 * :[ 'A4988', 'DRV8825', 'LV8729', 'L6470', 'TB6560', 'TB6600', 'TMC2100',
29 */
30 #define X_DRIVER_TYPE A4988 // comment out for stock drivers
31 #define Y_DRIVER_TYPE A4998 // comment out for stock drivers
32 #define Z_DRIVER_TYPE A4998 // comment out for stock drivers

```

```

1 #define X2_DRIVER_TYPE A4998
2 #define Y2_DRIVER_TYPE A4998
3 #define Z2_DRIVER_TYPE A4998 // comment out for stock drivers
4 #define E0_DRIVER_TYPE A4998 // comment out for stock drivers
5 #define E1_DRIVER_TYPE A4998 // comment out for stock drivers
6 #define E2_DRIVER_TYPE A4998
7 #define E3_DRIVER_TYPE A4998
8 #define E4_DRIVER_TYPE A4998

9
10 // Enable this feature if all enabled endstop pins are interrupt-capable.
11 // This will remove the need to poll the interrupt pins, saving many CPU cy
12 // #define ENDSTOP_INTERRUPTS_FEATURE

13
14 /**
15 * Endstop Noise Filter
16 *
17 * Enable this option if endstops falsely trigger due to noise.
18 * NOTE: Enabling this feature means adds an error of +/-0.2mm, so homing
19 * will end up at a slightly different position on each G28. This will also
20 * reduce accuracy of some bed probes.
21 * For mechanical switches, the better approach to reduce noise is to insta
22 * a 100 nanofarads ceramic capacitor in parallel with the switch, making it
23 * essentially noise-proof without sacrificing accuracy.
24 * This option also increases MCU load when endstops or the probe are enable
25 * So this is not recommended. USE AT YOUR OWN RISK.
26 * (This feature is not required for common micro-switches mounted on PCBs
27 * based on the Makerbot design, since they already include the 100nF capa
28 */
29 #define ENDSTOP_NOISE_FILTER

30
31 //=====
32 //===== Movement Settings =====

```

```

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

```

```

1 #define DEFAULT_MAX_FEEDRATE           { 500, 500, 6, 60 }

2

3 /**
4  * Default Max Acceleration (change/s) change = mm/s
5  * (Maximum start speed for accelerated moves)
6  * Override with M201
7  *
8 */
9 #define DEFAULT_MAX_ACCELERATION      { 3000, 2000, 60, 10000 }

10

11 /**
12  * Default Acceleration (change/s) change = mm/s
13  * Override with M204
14  *
15  * M204 P    Acceleration
16  * M204 R    Retract Acceleration
17  * M204 T    Travel Acceleration
18 */
19 #define DEFAULT_ACCELERATION          1500 // X, Y, Z and E acceleration
20 #define DEFAULT_RETRACT_ACCELERATION 3000 // E acceleration for retract
21 #define DEFAULT_TRAVEL_ACCELERATION   3000 // X, Y, Z acceleration for travel

22

23 /**
24  * Default Jerk (mm/s)
25  * Override with M205 X Y Z E
26  *
27  * "Jerk" specifies the minimum speed change that requires acceleration.
28  * When changing speed and direction, if the difference is less than the
29  * value set here, it may happen instantaneously.
30 */
31 #define DEFAULT_XJERK                10.0
32 #define DEFAULT_YJERK                10.0

```

```

1 #define DEFAULT_ZJERK 0.4
2 #define DEFAULT_EJERK 5.0
3
4 /**
5  * S-Curve Acceleration
6  *
7  * This option eliminates vibration during printing by fitting a Bezier
8  * curve to move acceleration , producing much smoother direction changes.
9  *
10 * See https://github.com/synthetos/TinyG/wiki/Jerk-Controlled-Motion-Explained
11 */
12 #define S_CURVE_ACCELERATION
13
14 //=====
15 //===== Z Probe Options =====
16 //=====
17 // @section probes
18
19 //
20 // See http://marlinfw.org/docs/configuration/probes.html
21 //
22
23 /**
24  * Z_MIN_PROBEUSES_Z_MIN_ENDSTOP_PIN
25  *
26  * Enable this option for a probe connected to the Z Min endstop pin .
27  */
28 // #define Z_MIN_PROBEUSES_Z_MIN_ENDSTOP_PIN
29
30 /**
31  * Z_MIN_PROBE_ENDSTOP
32  *

```

```

1   * Enable this option for a probe connected to any pin except Z-Min.
2   * (By default Marlin assumes the Z-Max endstop pin.)
3   * To use a custom Z Probe pin, set Z_MIN_PROBE_PIN below.
4   *
5   * -- The simplest option is to use a free endstop connector.
6   * -- Use 5V for powered (usually inductive) sensors.
7   *
8   * -- RAMPs 1.3/1.4 boards may use the 5V, GND, and Aux4->D32 pin:
9   *     -- For simple switches connect...
10  *         -- normally-closed switches to GND and D32.
11  *         -- normally-open switches to 5V and D32.
12  *
13  * WARNING: Setting the wrong pin may have unexpected and potentially
14  * disastrous consequences. Use with caution and do your homework.
15  *
16  */
17 #define Z_MIN_PROBE_ENDSTOP
18
19 /**
20  * Probe Type
21  *
22  * Allen Key Probes, Servo Probes, Z-Sled Probes, FIX_MOUNTED_PROBE, etc.
23  * Activate one of these to use Auto Bed Leveling below.
24  */
25
26 /**
27  * The "Manual Probe" provides a means to do "Auto" Bed Leveling without a
28  * Use G29 repeatedly, adjusting the Z height at each point with movement
29  * or (with LCD_BED_LEVELING) the LCD controller.
30  */
31 #define PROBE_MANUALLY
32 // #define MANUAL_PROBE_START_Z 0.2

```

```

1
2  /**
3   * A Fix-Mounted Probe either doesn't deploy or needs manual deployment.
4   * (e.g., an inductive probe or a nozzle-based probe-switch.)
5   */
6 // #define FIX_MOUNTED_PROBE
7
8 /**
9  * Z Servo Probe, such as an endstop switch on a rotating arm.
10 */
11 // #define Z_PROBE_SERVO_NR 0      // Defaults to SERVO 0 connector.
12 // #define Z_SERVO_ANGLES {70,0}    // Z Servo Deploy and Stow angles
13
14 /**
15  * The BLTouch probe uses a Hall effect sensor and emulates a servo.
16 */
17 // #define BLTOUCH
18
19 /**
20  * Enable one or more of the following if probing seems unreliable.
21  * Heaters and/or fans can be disabled during probing to minimize electrical
22  * noise. A delay can also be added to allow noise and vibration to settle.
23  * These options are most useful for the BLTouch probe, but may also improve
24  * readings with inductive probes and piezo sensors.
25 */
26 // #define PROBING_HEATERS_OFF          // Turn heaters off when probing
27 #if ENABLED(PROBING_HEATERS_OFF)
28     // #define WAIT_FOR_BED_HEATER        // Wait for bed to heat back up between
29 #endif
30 // #define PROBING_FANS_OFF           // Turn fans off when probing
31 // #define DELAY_BEFORE_PROBING 200    // (ms) To prevent vibrations from trig
32

```

```

1 // A probe that is deployed and stowed with a solenoid pin (SOL1_PIN)
2 // #define SOLENOID_PROBE
3
4 // A sled-mounted probe like those designed by Charles Bell.
5 // #define Z_PROBE_SLED
6 // #define SLED_DOCKING_OFFSET 5 // The extra distance the X axis must travel
7
8 //
9 // For Z_PROBE_ALLEN_KEY see the Delta example configurations.
10 //
11
12 /**
13 * Z Probe to nozzle (X,Y) offset , relative to (0, 0).
14 * X and Y offsets must be integers .
15 *
16 * In the following example the X and Y offsets are both positive:
17 * #define X_PROBE_OFFSET_FROM_EXTRUDER 10
18 * #define Y_PROBE_OFFSET_FROM_EXTRUDER 10
19 *
20 *      +--- BACK ---+
21 *      |           |
22 *      L | (+) P | R <-- probe (20,20)
23 *      E |           | I
24 *      F | (-) N (+) | G <-- nozzle (10,10)
25 *      T |           | H
26 *      | (-)       | T
27 *      |           |
28 *      O— FRONT —+
29 *      (0,0)
30 */
31 #define X_PROBE_OFFSET_FROM_EXTRUDER 10 // X offset : -left +right
32 [ of the nozzle ]

```

```

1 #define Y_PROBE_OFFSET_FROM_EXTRUDER 10 // Y offset: -front +behind [ the
2 #define Z_PROBE_OFFSET_FROM_EXTRUDER 0 // Z offset: -below +above
3 [ the nozzle]
4
5 // Certain types of probes need to stay away from edges
6 #define MIN_PROBE_EDGE 10
7
8 // X and Y axis travel speed (mm/m) between probes
9 #define XY_PROBE_SPEED 8000
10
11 // Feedrate (mm/m) for the first approach when double-probing (MULTIPLE_PROBING)
12 #define Z_PROBE_SPEED_FAST HOMING_FEEDRATE_Z
13
14 // Feedrate (mm/m) for the "accurate" probe of each point
15 #define Z_PROBE_SPEED_SLOW (Z_PROBE_SPEED_FAST / 2)
16
17 // The number of probes to perform at each point.
18 // Set to 2 for a fast/slow probe, using the second probe result.
19 // Set to 3 or more for slow probes, averaging the results.
20 // #define MULTIPLE_PROBING 2
21
22 /**
23 * Z probes require clearance when deploying, stowing, and moving between
24 * probe points to avoid hitting the bed and other hardware.
25 * Servo-mounted probes require extra space for the arm to rotate.
26 * Inductive probes need space to keep from triggering early.
27 *
28 * Use these settings to specify the distance (mm) to raise the probe (or
29 * lower the bed). The values set here apply over and above any (negative)
30 * probe Z Offset set with Z_PROBE_OFFSET_FROM_EXTRUDER, M851, or the LCD.
31 * Only integer values >= 1 are valid here.
32 */

```

```

1   * Example: 'M851 Z-5' with a CLEARANCE of 4 => 9mm from bed to nozzle.
2   * But: 'M851 Z+1' with a CLEARANCE of 2 => 2mm from bed to nozzle.
3   */
4 #define Z_CLEARANCE_DEPLOY_PROBE    10 // Z Clearance for Deploy/Stow
5 #define Z_CLEARANCE_BETWEEN_PROBES  5 // Z Clearance between probe points
6 #define Z_CLEARANCE_MULTIPROBE     5 // Z Clearance between multiple probe
7 // #define Z_AFTER_PROBING          5 // Z position after probing is done
8
9 #define Z_PROBE_LOW_POINT          -2 // Farthest distance below the trigger
10
11 // For M851 give a range for adjusting the Z probe offset
12 #define Z_PROBE_OFFSET_RANGE_MIN -20
13 #define Z_PROBE_OFFSET_RANGE_MAX  20
14
15 // Enable the M48 repeatability test to test probe accuracy
16 // #define Z_MIN_PROBE_REPEATABILITY_TEST
17
18 // For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Act
19 // :{ 0:'Low', 1:'High' }
20 #define X_ENABLE_ON 0
21 #define Y_ENABLE_ON 0
22 #define Z_ENABLE_ON 0
23 #define E_ENABLE_ON 0 // For all extruders
24
25 // Disables axis stepper immediately when it's not being used.
26 // WARNING: When motors turn off there is a chance of losing position accu
27 #define DISABLE_X false
28 #define DISABLE_Y false
29 #define DISABLE_Z false
30 // Warn on display about possibly reduced accuracy
31 // #define DISABLE_REDUCED_ACCURACY_WARNING
32

```

```

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

```

```

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

```

```

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

```
1 //===== Bed Leveling =====
2 //=====
3 // @section calibrate
4
5 /**
6 * Choose one of the options below to enable G29 Bed Leveling. The parameters
7 * and behavior of G29 will change depending on your selection.
8 *
9 * If using a Probe for Z Homing, enable Z_SAFE_HOMING also!
10 *
11 * — AUTO_BED_LEVELING_3POINT
12 * Probe 3 arbitrary points on the bed (that aren't collinear)
13 * You specify the XY coordinates of all 3 points.
14 * The result is a single tilted plane. Best for a flat bed.
15 *
16 * — AUTO_BED_LEVELING_LINEAR
17 * Probe several points in a grid.
18 * You specify the rectangle and the density of sample points.
19 * The result is a single tilted plane. Best for a flat bed.
20 *
21 * — AUTO_BED_LEVELING_BILINEAR
22 * Probe several points in a grid.
23 * You specify the rectangle and the density of sample points.
24 * The result is a mesh, best for large or uneven beds.
25 *
26 * — AUTO_BED_LEVELING_UBL (Unified Bed Leveling)
27 * A comprehensive bed leveling system combining the features and benefits
28 * of other systems. UBL also includes integrated Mesh Generation, Mesh
29 * Validation and Mesh Editing systems.
30 *
31 * — MESH_BED_LEVELING
32 * Probe a grid manually
```

```

1  *      The result is a mesh, suitable for large or uneven beds. (See BILINEAR)
2  *      For machines without a probe, Mesh Bed Leveling provides a method to p
3  *      leveling in steps so you can manually adjust the Z height at each grid
4  *      With an LCD controller the process is guided step-by-step.
5  */
6 // #define AUTO_BED_LEVELING_3POINT
7 // #define AUTO_BED_LEVELING_LINEAR
8 // #define AUTO_BED_LEVELING_BILINEAR
9 // #define AUTO_BED_LEVELING_UBL
10 #define MESH_BED_LEVELING
11
12 /**
13 * Normally G28 leaves leveling disabled on completion. Enable
14 * this option to have G28 restore the prior leveling state.
15 */
16 // #define RESTORE_LEVELING_AFTER_G28
17
18 /**
19 * Enable detailed logging of G28, G29, M48, etc.
20 * Turn on with the command 'M111 S32'.
21 * NOTE: Requires a lot of PROGMEM!
22 */
23 // #define DEBUG_LEVELING_FEATURE
24
25 #if ENABLED(MESH_BED_LEVELING) || ENABLED(AUTO_BED_LEVELING_BILINEAR) || EN
26 // Gradually reduce leveling correction until a set height is reached,
27 // at which point movement will be level to the machine's XY plane.
28 // The height can be set with M420 Z<height>
29 #define ENABLE_LEVELING_FADE_HEIGHT
30
31 // For Cartesian machines, instead of dividing moves on mesh boundaries,
32 // split up moves into short segments like a Delta. This follows the

```

```

1 // contours of the bed more closely than edge-to-edge straight moves.
2 #define SEGMENT_LEVELED_MOVES
3 #define LEVELED_SEGMENT_LENGTH 5.0 // (mm) Length of all segments (except
4
5 /**
6 * Enable the G26 Mesh Validation Pattern tool.
7 */
8 #define G26_MESH_VALIDATION
9 #if ENABLED(G26_MESH_VALIDATION)
10 #define MESH_TEST_NOZZLE_SIZE      0.4 // (mm) Diameter of primary nozzle
11 #define MESH_TEST_LAYER_HEIGHT    0.2 // (mm) Default layer height for
12 #define MESH_TEST_HOTEND_TEMP    200.0 // ° (C) Default nozzle temperature
13 #define MESH_TEST_BED_TEMP       60.0 // ° (C) Default bed temperature
14#endif
15
16#endif
17
18#if ENABLED(AUTO_BED_LEVELING_LINEAR) || ENABLED(AUTO_BED_LEVELING_BILINEAR)
19
20 // Set the number of grid points per dimension.
21 #define GRID_MAX_POINTS_X 3
22 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
23
24 // Set the boundaries for probing (where the probe can reach).
25 #define LEFT_PROBE_BED_POSITION 25
26 #define RIGHT_PROBE_BED_POSITION 181
27 #define FRONT_PROBE_BED_POSITION 25
28 #define BACK_PROBE_BED_POSITION 185
29
30 // The Z probe minimum outer margin (to validate G29 parameters).
31 #define MIN_PROBE_EDGE 10
32

```

```

1 // Probe along the Y axis , advancing X after each column
2 //">#define PROBE_Y_FIRST
3
4 #if ENABLED(AUTO_BED_LEVELING_BILINEAR)
5
6 // Beyond the probed grid , continue the implied tilt ?
7 // Default is to maintain the height of the nearest edge.
8 //">#define EXTRAPOLATE_BEYOND_GRID
9
10 //
11 // Experimental Subdivision of the grid by Catmull-Rom method .
12 // Synthesizes intermediate points to produce a more detailed mesh .
13 //
14 //">#define ABL_BILINEAR_SUBDIVISION
15 #if ENABLED(ABL_BILINEAR_SUBDIVISION)
16 // Number of subdivisions between probe points
17 #define BILINEAR_SUBDIVISIONS 3
18#endif
19
20#endif
21
22#elif ENABLED(AUTO_BED_LEVELING_UBL)
23
24 //=====
25 //===== Unified Bed Leveling =====
26 //=====
27
28 //">#define MESH_EDIT_GFX_OVERLAY // Display a graphics overlay while edi
29
30#define MESH_INSET 1 // Set Mesh bounds as an inset region of the print
31#define GRID_MAX_POINTS_X 10 // Don't use more than 15 points per axis
32#define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

```

```

1
2 #define UBL_MESH_EDIT_MOVES_Z      // Sophisticated users prefer no movement
3 #define UBL_SAVE_ACTIVE_ON_M500   // Save the currently active mesh in the
4
5 // #define UBL_Z_RAISE_WHEN_OFF_MESH 2.5 // When the nozzle is off the mesh
6 //                                         // as the Z-Height correction value
7
8 #elif ENABLED(MESH_BED_LEVELING)
9
10 //=====
11 //===== Mesh =====
12 //=====
13
14 #define MESH_INSET 10           // Set Mesh bounds as an inset region of the
15 #define GRID_MAX_POINTS_X 5     // Don't use more than 7 points per axis,
16 #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X
17
18 // #define MESH_G28_REST_ORIGIN // After homing all axes ('G28' or 'G28 XY')
19
20 #endif // BED_LEVELING
21
22 /**
23 * Points to probe for all 3-point Leveling procedures.
24 * Override if the automatically selected points are inadequate.
25 */
26 #if ENABLED(AUTO_BED_LEVELING_3POINT) || ENABLED(AUTO_BED_LEVELING_UBL)
27 // #define PROBE_PT_1_X 15
28 // #define PROBE_PT_1_Y 180
29 // #define PROBE_PT_2_X 15
30 // #define PROBE_PT_2_Y 20
31 // #define PROBE_PT_3_X 170
32 // #define PROBE_PT_3_Y 20

```

```

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

```

```

1 // #define BED_CENTER_AT_0_0
2
3 // Manually set the home position. Leave these undefined for automatic sett
4 // For DELTA this is the top-center of the Cartesian print volume.
5 // #define MANUAL_X_HOME_POS 0
6 // #define MANUAL_Y_HOME_POS 0
7 // #define MANUAL_Z_HOME_POS 0
8
9 // Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
10 //
11 // With this feature enabled:
12 //
13 // - Allow Z homing only after X and Y homing AND stepper drivers still ena
14 // - If stepper drivers time out, it will need X and Y homing again before
15 // - Move the Z probe (or nozzle) to a defined XY point before Z Homing whe
16 // - Prevent Z homing when the Z probe is outside bed area.
17 //
18 // #define Z_SAFE_HOMING
19
20 #if ENABLED(Z_SAFE_HOMING)
21     #define Z_SAFE_HOMING_X_POINT ((X_BED_SIZE) / 2)      // X point for Z hom
22     #define Z_SAFE_HOMING_Y_POINT ((Y_BED_SIZE) / 2)      // Y point for Z hom
23 #endif
24
25 // Homing speeds (mm/m)
26 #define HOMING_FEEDRATE_XY (50*60)
27 #define HOMING_FEEDRATE_Z (4*60)
28
29 // @section calibrate
30
31 /**
32 * Bed Skew Compensation

```

```

1   *
2   * This feature corrects for misalignment in the XYZ axes.
3   *
4   * Take the following steps to get the bed skew in the XY plane:
5   * 1. Print a test square (e.g., https://www.thingiverse.com/thing:2563185)
6   * 2. For XY_DIAG_AC measure the diagonal A to C
7   * 3. For XY_DIAG_BD measure the diagonal B to D
8   * 4. For XY_SIDE_AD measure the edge A to D
9   *
10  * Marlin automatically computes skew factors from these measurements.
11  * Skew factors may also be computed and set manually:
12  *
13  * - Compute AB      :  $\text{SQRT}(2*\text{AC}*\text{AC}+2*\text{BD}*\text{BD}-4*\text{AD}*\text{AD})/2$ 
14  * - XY_SKEW_FACTOR :  $\text{TAN}(\text{PI}/2-\text{ACOS}((\text{AC}*\text{AC}-\text{AB}*\text{AB}-\text{AD}*\text{AD})/(2*\text{AB}*\text{AD})))$ 
15  *
16  * If desired, follow the same procedure for XZ and YZ.
17  * Use these diagrams for reference:
18  *
19  *      Y          Z          Z
20  *      ^     B---C      ^     B---C      ^     B---C
21  *      |     /     /      |     /     /      |     /     /
22  *      |     /     /      |     /     /      |     /     /
23  *      |     A---D      |     A---D      |     A---D
24  *      +----->X      +----->X      +----->Y
25  *      XY_SKEW_FACTOR      XZ_SKEW_FACTOR      YZ_SKEW_FACTOR
26  */
27 // #define SKEW_CORRECTION
28
29 #if ENABLED(SKEW_CORRECTION)
30   // Input all length measurements here:
31   #define XY_DIAG_AC 282.8427124746
32   #define XY_DIAG_BD 282.8427124746

```

```

1 #define XY_SIDE_AD 200
2
3 // Or, set the default skew factors directly here
4 // to override the above measurements:
5 #define XY_SKew_FACTOR 0.0
6
7 // #define SKEW_CORRECTION_FOR_Z
8 #if ENABLED(SKEW_CORRECTION_FOR_Z)
9     #define XZ_DIAG_AC 282.8427124746
10    #define XZ_DIAG_BD 282.8427124746
11    #define YZ_DIAG_AC 282.8427124746
12    #define YZ_DIAG_BD 282.8427124746
13    #define YZ_SIDE_AD 200
14    #define XZ_SKew_FACTOR 0.0
15    #define YZ_SKew_FACTOR 0.0
16 #endif
17
18 // Enable this option for M852 to set skew at runtime
19 // #define SKEW_CORRECTION_GCODE
20#endif
21
22 //=====
23 //===== Additional Features =====
24 //=====
25
26 // @section extras
27
28 //
29 // EEPROM
30 //
31 // The microcontroller can store settings in the EEPROM, e.g. max velocity
32 // M500 — stores parameters in EEPROM

```

```

1 // M501 — reads parameters from EEPROM (if you need reset them after you change
2 // M502 — reverts to the default "factory settings". You still need to store them
3 //
4 #define EEPROMSETTINGS // Enable for M500 and M501 commands
5 // #define DISABLE_M503 // Saves ~2700 bytes of PROGMEM. Disable for reliability
6 #define EEPROMCHITCHAT // Give feedback on EEPROM commands. Disable to save memory
7 //
8 //
9 // Host Keepalive
10 //
11 // When enabled Marlin will send a busy status message to the host
12 // every couple of seconds when it can't accept commands.
13 //
14 #define HOST_KEEPALIVE_FEATURE // Disable this if your host doesn't support it
15 #define DEFAULT_KEEPALIVE_INTERVAL 2 // Number of seconds between "busy" messages
16 #define BUSY_WHILE_HEATING // Some hosts require "busy" messages during heating
17 //
18 //
19 // M100 Free Memory Watcher
20 //
21 // #define M100_FREE_MEMORY_WATCHER // Add M100 (Free Memory Watcher) to G-code
22 //
23 //
24 // G20/G21 Inch mode support
25 //
26 // #define INCH_MODE_SUPPORT
27 //
28 //
29 // M149 Set temperature units support
30 //
31 // #define TEMPERATURE_UNITS_SUPPORT
32

```

```

1 // @section temperature
2
3 // Preheat Constants
4 #define PREHEAT_1_TEMP_HOTEND 180
5 #define PREHEAT_1_TEMP_BED      70
6 #define PREHEAT_1_FAN_SPEED    0 // Value from 0 to 255
7
8 #define PREHEAT_2_TEMP_HOTEND 240
9 #define PREHEAT_2_TEMP_BED      110
10 #define PREHEAT_2_FAN_SPEED   0 // Value from 0 to 255
11
12 /**
13 * Nozzle Park
14 *
15 * Park the nozzle at the given XYZ position on idle or G27.
16 *
17 * The "P" parameter controls the action applied to the Z axis:
18 *
19 *     P0 (Default) If Z is below park Z raise the nozzle.
20 *     P1 Raise the nozzle always to Z-park height.
21 *     P2 Raise the nozzle by Z-park amount, limited to Z_MAX_POS.
22 */
23 #define NOZZLE_PARK_FEATURE
24
25 #if ENABLED(NOZZLE_PARK_FEATURE)
26     // Specify a park position as { X, Y, Z }
27     #define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }
28     #define NOZZLE_PARK_XY_FEEDRATE 100 // X and Y axes feedrate in mm/s (a
29     #define NOZZLE_PARK_Z_FEEDRATE 5      // Z axis feedrate in mm/s (not use
30 #endif
31
32 /**

```

```

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

```

```

1   *
2   *      Caveats: The ending Z should be the same as starting Z.
3   * Attention: EXPERIMENTAL. G-code arguments may change.
4   *
5   */
6 // #define NOZZLE_CLEAN_FEATURE
7
8 #if ENABLED(NOZZLE_CLEAN_FEATURE)
9   // Default number of pattern repetitions
10  #define NOZZLE_CLEAN_STROKES 12
11
12  // Default number of triangles
13  #define NOZZLE_CLEAN_TRIANGLES 3
14
15  // Specify positions as { X, Y, Z }
16  #define NOZZLE_CLEAN_START_POINT { 30, 30, (Z_MIN_POS + 1) }
17  #define NOZZLE_CLEAN_END_POINT { 100, 60, (Z_MIN_POS + 1) }
18
19  // Circular pattern radius
20  #define NOZZLE_CLEAN_CIRCLE_RADIUS 6.5
21  // Circular pattern circle fragments number
22  #define NOZZLE_CLEAN_CIRCLE_FN 10
23  // Middle point of circle
24  #define NOZZLE_CLEAN_CIRCLE_MIDDLE NOZZLE_CLEAN_START_POINT
25
26  // Moves the nozzle to the initial position
27  #define NOZZLE_CLEAN_GOBACK
28 #endif
29
30 /**
31  * Print Job Timer
32  *

```

```

1  * Automatically start and stop the print job timer on M104/M109/M190.
2  *
3  *      M104 (hotend, no wait) — high temp = none,           low temp = stop time
4  *      M109 (hotend, wait)    — high temp = start timer, low temp = stop time
5  *      M190 (bed, wait)      — high temp = start timer, low temp = none
6  *
7  * The timer can also be controlled with the following commands:
8  *
9  *      M75 — Start the print job timer
10 *     M76 — Pause the print job timer
11 *     M77 — Stop the print job timer
12 */
13 #define PRINTJOB_TIMER_AUTOSTART
14
15 /**
16  * Print Counter
17 *
18  * Track statistical data such as:
19 *
20  * — Total print jobs
21  * — Total successful print jobs
22  * — Total failed print jobs
23  * — Total time printing
24 *
25  * View the current statistics with M78.
26 */
27 #define PRINTCOUNTER
28
29 //=====
30 //===== LCD and SD support =====
31 //=====
32

```

```

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

```

```

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

```

```

1
2  /**
3   * LCD Menu Items
4   *
5   * Disable all menus and only display the Status Screen , or
6   * just remove some extraneous menu items to recover space .
7   */
8 // #define NO_LCD_MENUS
9 // #define SLIM_LCD_MENUS

10
11 //
12 // ENCODER SETTINGS
13 //
14 // This option overrides the default number of encoder pulses needed to
15 // produce one step . Should be increased for high-resolution encoders .
16 //
17 // #define ENCODER_PULSES_PER_STEP 4

18
19 //
20 // Use this option to override the number of step signals required to
21 // move between next/prev menu items .
22 //
23 // #define ENCODER_STEPS_PER_MENU_ITEM 1

24
25 /**
26   * Encoder Direction Options
27   *
28   * Test your encoder 's behavior first with both options disabled .
29   *
30   * Reversed Value Edit and Menu Nav? Enable REVERSE_ENCODER_DIRECTION .
31   * Reversed Menu Navigation only?      Enable REVERSE_MENU_DIRECTION .
32   * Reversed Value Editing only?       Enable BOTH options .

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

1 // a pin (if BTN_ENC defined to != -1) or read through I2C (when BTN_ENC == -1)
2 //
3 // #define LCD_I2C_PANELOLU2
4
5 //
6 // Panucatt VIKI LCD with status LEDs,
7 // integrated click & L/R/U/D buttons , separate encoder inputs .
8 //
9 // #define LCD_I2C_VIKI
10
11 //
12 // CONTROLLER TYPE: Shift register panels
13 //
14
15 //
16 // 2 wire Non-latching LCD SR from https://goo.gl/aJJ4sH
17 // LCD configuration : http://reprap.org/wiki/SAV_3D_LCD
18 //
19 // #define SAV_3DLCD
20
21 //=====
22 //===== LCD / Controller Selection =====
23 //===== (Graphical LCDs) =====
24 //=====
25
26 //
27 // CONTROLLER TYPE: Graphical 128x64 (DOGM)
28 //
29 // IMPORTANT: The U8glib library is required for Graphical Display !
30 // https://github.com/olikraus/U8glib_Arduino
31 //
32

```

```

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

```

```

1  // 
2  // Adafruit ST7565 Full Graphic Controller .
3  // https://github.com/eboston/Adafruit-ST7565-Full-Graphic-Controller/
4  //
5  // #define ELB_FULL_GRAPHIC_CONTROLLER
6
7  //
8  // BQ LCD Smart Controller shipped by
9  // default with the BQ Hephestos 2 and Witbox 2 .
10 //
11 // #define BQLCDSMARTCONTROLLER
12
13 //
14 // Cartesio UI
15 // http://mauk.cc/webshop/cartesio-shop/electronics/user-interface
16 //
17 // #define CARTESIO_UI
18
19 //
20 // LCD for Melzi Card with Graphical LCD
21 //
22 // #define LCD_FOR_MELZI
23
24 //
25 // SSD1306 OLED full graphics generic display
26 //
27 // #define U8GLIB_SSD1306
28
29 //
30 // SAV OLED LCD module support using either SSD1306 or SH1106 based LCD mo
31 //
32 // #define SAV_3DGLCD

```

```

1 #if ENABLED(SAV_3DGLCD)
2 // #define U8GLIB_SSD1306
3 #define U8GLIB_SH1106
4 #endif
5
6 //
7 // Original Ulticontroller from Ultimaker 2 printer with SSD1309 I2C display
8 // https://github.com/Ultimaker/Ultimaker2/tree/master/1249_Ulticontroller
9 //
10 // #define ULTI_CONTROLLER
11
12 //
13 // TinyBoy2 128x64 OLED / Encoder Panel
14 //
15 // #define OLED_PANEL_TINYBOY2
16
17 //
18 // MKS MINI12864 with graphic controller and SD support
19 // http://reprap.org/wiki/MKS_MINI_12864
20 //
21 // #define MKS_MINI_12864
22
23 //
24 // Factory display for Creality CR-10
25 // https://www.aliexpress.com/item/Universal-LCD-12864-3D-Printer-Display-S
26 //
27 // This is RAMPS-compatible using a single 10-pin connector.
28 // (For CR-10 owners who want to replace the Melzi Creality board but retain
29 //
30 // #define CR10_STOCKDISPLAY
31
32 //

```

```

1 // ANET and Tronxy Graphical Controller
2 //
3 // #define ANET_FULL_GRAPHICS_LCD // Anet 128x64 full graphics lcd with rot
4 // A clone of the RepRapDiscount full gra
5 // different pins/wiring (see pins_ANET-1
6
7 //
8 // MKS OLED 1.3" 128 × 64 FULL GRAPHICS CONTROLLER
9 // http://reprap.org/wiki/MKS_12864OLED
10 //
11 // Tiny , but very sharp OLED display
12 //
13 // #define MKS_12864OLED // Uses the SH1106 controller (default)
14 // #define MKS_12864OLED_SSD1306 // Uses the SSD1306 controller
15
16 //
17 // Silvergate GLCD controller
18 // http://github.com/android444/Silvergate
19 //
20 // #define SILVER_GATE_GLCD_CONTROLLER
21
22 //=====
23 //===== Other Controllers =====
24 //=====
25
26 //
27 // CONTROLLER TYPE: Standalone / Serial
28 //
29
30 //
31 // LCD for Malyan M200 printers .
32 // This requires SDSUPPORT to be enabled

```

```

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

```

```

1 // affecting heaters , and the fan if FANSOFTPWM is enabled .
2 // However, control resolution will be halved for each increment ;
3 // at zero value , there are 128 effective control positions .
4 #define SOFT_PWMSCALE 0

5

6 // If SOFT_PWMSCALE is set to a value higher than 0, dithering can
7 // be used to mitigate the associated resolution loss. If enabled ,
8 // some of the PWM cycles are stretched so on average the desired
9 // duty cycle is attained .

10 // #define SOFT_PWM_DITHER

11

12 // Temperature status LEDs that display the hotend and bed temperature .
13 // If all hotends , bed temperature , and target temperature are under 54C
14 // then the BLUE led is on. Otherwise the RED led is on. (1C hysteresis)
15 // #define TEMP_STAT_LEDS

16

17 // M240 Triggers a camera by emulating a Canon RC-1 Remote
18 // Data from: http://www.doc-diy.net/photo/rc-1-hacked/
19 // #define PHOTOGRAPH_PIN 23

20

21 // SkeinForge sends the wrong arc g-codes when using Arc Point as fillet p
22 // #define SF_ARC_FIX

23

24 // Support for the BariCUDA Paste Extruder
25 // #define BARICUDA

26

27 // Support for BlinkM/CyzRgb
28 // #define BLINKM

29

30 // Support for PCA9632 PWM LED driver
31 // #define PCA9632

32

```

```

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

```

```

1 // Support for Adafruit Neopixel LED driver
2 // #define NEOPIXELLED
3 #if ENABLED(NEOPIXELLED)
4     #define NEOPIXEL_TYPE    NEO_GRBW // NEO_GRBW / NEO_GRB – four/three channel
5     #define NEOPIXEL_PIN      4          // LED driving pin on motherboard 4 => I2C
6     #define NEOPIXEL_PIXELS   30         // Number of LEDs in the strip
7     #define NEOPIXEL_IS_SEQUENTIAL // Sequential display for temperature change
8     #define NEOPIXEL_BRIGHTNESS 127 // Initial brightness (0–255)
9     // #define NEOPIXEL_STARTUP_TEST // Cycle through colors at startup
10 #endif
11
12 /**
13 * Printer Event LEDs
14 *
15 * During printing , the LEDs will reflect the printer status:
16 *
17 * – Gradually change from blue to violet as the heated bed gets to target
18 * – Gradually change from violet to red as the hotend gets to temperature
19 * – Change to white to illuminate work surface
20 * – Change to green once print has finished
21 * – Turn off after the print has finished and the user has pushed a button
22 */
23 #if ENABLED(BLINKM) || ENABLED(RGB_LED) || ENABLED(RGBW_LED) || ENABLED(PCF8574)
24     #define PRINTER_EVENT_LEDS
25 #endif
26
27 /**
28 * R/C SERVO support
29 * Sponsored by TrinityLabs , Reworked by codexmas
30 */
31
32 /**

```

```

1  * Number of servos
2  *
3  * For some servo-related options NUMSERVOS will be set automatically.
4  * Set this manually if there are extra servos needing manual control.
5  * Leave undefined or set to 0 to entirely disable the servo subsystem.
6  */
7 // #define NUMSERVOS 3 // Servo index starts with 0 for M280 command
8
9 // Delay (in milliseconds) before the next move will start , to give the ser
10 // 300ms is a good value but you can try less delay.
11 // If the servo can't reach the requested position , increase it .
12 #define SERVO_DELAY { 300 }

13
14 // Only power servos during movement, otherwise leave off to prevent jitter
15 // #define DEACTIVATESERVOSAFTERMOVE

16
17 /**
18 * Select your version of the Trigorilla (RAMPS1.4) board here .
19 *
20 * 0 = Default Trigorilla
21 * 1 = Newer Trigorilla v1.1 (first seen late 2018)
22 *
23 * The only major difference is a slight change on the servo pin mapping.
24 * This setting only is relevant if you want to use BLtouch or similar
25 * mods to be used via servo pins .
26 * The new version is to be identified by a "TRIGORILLA1.1" lettering
27 * on the upper left of the PCB silkscreen .
28 */
29 #define TRIGORILLA_VERSION 0
30
31 // Enable Anycubic TFT
32 #define ANYCUBIC_TFT_MODEL

```

```
1 #define ANYCUBIC_FILAMENT_RUNOUT_SENSOR
2 // #define ANYCUBIC_TFT_DEBUG
3
4 #endif // CONFIGURATION.H
```