

Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
_____ О.В. Непомнящий
" ____ " _____ 2023 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Аппаратно-программный комплекс для реализации физических
3D-моделей. Программная часть

Руководитель	_____	_____	доцент, канд. техн. наук В.Т.	В.Г. Середкин
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____	ЗКИ18 – 07Б, 031205722	В.В. Юшков
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____		В.Г. Середкин
	<i>подпись</i>	<i>дата</i>		

Красноярск 2023

РЕФЕРАТ

Выпускная квалификационная работа на тему «Аппаратно-программный комплекс для реализации физических 3D-моделей. Программная часть. Содержит 66 страниц, 35 рисунков, 5 таблиц, 15 источников.

ОТКРЫТЫЙ ПРОЕКТ, АГРЕГАТ, 3D-КОМПЛЕКС, СРЕДА РАЗРАБОТКИ, ПРОШИВКА, ARDUINO IDE, PRUSA MENDEL, MARLIN.

Целью данной выпускной квалификационной работы является разработка прототипа аппаратно-программного комплекса для реализации физических 3D-моделей на микроконтроллерной основе.

Поставленные задачи:

- анализ задания и обоснование выбранного решения;
- разработка аппаратно-программных моделей функциональных узлов;
- тестирование программного обеспечения для разрабатываемого комплекса.

В результате выпускной квалификационной работы был разработан прототип на основе открытого проекта Prusa Mendel.

Практическая значимость состоит в том, что она даёт возможность реализовывать физические 3D-модели высокого разрешения и при высокой скорости печати, что можно применять в различных отраслях промышленности, а также в учебных целях.

СОДЕРЖАНИЕ

Введение	5
1 Анализ задания и обоснование принимаемого решения	6
1.1 Обзор источников информации по предметной области.....	6
1.2 Анализ существующих аналогов 3D-комплексов	7
1.2.1 Технические характеристики	8
1.2.2 Преимущества и недостатки 3D-комплексов	10
1.3 Анализ и выбор программно-аппаратного обеспечения системного назначения	12
1.3.1 Открытый проект «Teacup»	13
1.3.2 Открытый проект «Sprinter»	14
1.3.3 Среда разработки PlatformIO.....	15
1.3.4 Среда разработки Arduino Eclipse plugin	16
1.3.5 Платформа на базе Arduino Mega 2560	17
1.3.6 Плата расширения (Shield) Ramps 1.4	18
1.4 Обоснование выбора принимаемого решения	21
1.4.1 Система Prusa Mendel.....	21
1.4.2 Среда разработки Arduino IDE.....	25
1.4.3 Открытый проект «Marlin»	26
2 Описание объекта управления (электромеханического агрегата), разработка информационной модели процесса управления и аппаратно-программных моделей функциональных узлов разрабатываемой системы	28
2.1 Информационная модель процесса управления электромеханическим агрегатом.....	28
2.2 Особенности программирования функционального узла «материнская плата»	30
2.3 Программирование функционального узла «дисплейный модуль»	32
2.4 Пример программного подключения функционального узла «термистор»	34
2.5 Программные ограничения функционального узла «экструдер»	35

2.6 Программирование функционального узла «нагревательный стол»	37
2.7 Программирование параметров функционального узла «концевой переключатель»	38
2.8 Особенности программирования системы движения осей	39
2.8.1 Математическое обеспечение механизма движения осей	40
2.8.2 Программная настройка скорости подачи и ускорения	44
2.8.3 Программное управление инвертированием движения шаговых двигателей	45
2.8.4 Программное управление областью печати	45
2.9 Программное подключение функции распознавания SD-карт	48
3 Тестирование программного обеспечения для разрабатываемого комплекса	50
3.1.1 Программная калибровка PID-контроллера температур	52
3.1.2 Программно-инструментальная калибровка подачи филамента	54
3.1.3 Подготовка тестовой модели	58
3.1.4 Процесс реализации физической 3D-модели	62
Заключение	64
Список использованных источников	65

ВВЕДЕНИЕ

3D-печать становится все более распространенной и находит применение в различных сферах деятельности, что существенно расширяет её область применения в повседневной жизни людей. С помощью этой технологии создают трехмерную модель изделия на компьютере, которая может быть распечатана за короткое время, чтобы получить реальный предмет, точно соответствующий разработанной 3D-модели. 3D-печать позволяет производить множество объектов, в том числе уникальные и изготовленные по индивидуальному заказу продукты и детали.

Актуальность разработки аппаратно-программного комплекса и системы управления 3D-комплексом состоит в том, чтобы спроектировать и запрограммировать прототип не уступающий серийным комплексам, и сделать прототип более доступным для индивидуального использования, который позволит достичь более высокой эффективности и точности в работе с физическими 3D-моделями. Прототип позволяет упростить и автоматизировать процесс создания трехмерных моделей, управлять этапами печати, контролировать качество печати и управлять модернизированными функциональными узлами более эффективно, что можно применять в различных отраслях промышленности, а также в учебных целях.

Целью данной выпускной квалификационной работы является разработка прототипа аппаратно-программного комплекса для реализации физических 3D-моделей на микроконтроллерной основе.

Для достижения цели, ставим следующие задачи:

- анализ задания и обоснование выбранного решения;
- разработка аппаратно-программных моделей функциональных узлов прототипа;
- тестирование программного обеспечения для разрабатываемого комплекса.

1 Анализ задания и обоснование принимаемого решения

1.1 Обзор источников информации по предметной области

3D-печать появилась более 30 лет назад, когда основатель компании 3D Systems Чарльз Халл (Charles Hull) в 1986 году разработал первый 3D-принтер и изобрел технологию stereolithography. Она основывалась на использовании лазера для затвердевания жидкой смолы и предназначалась для быстрого создания небольших прототипов трехмерных моделей.

В течение последующих лет были созданы другие технологии 3D-печати, включая fused deposition modeling (FDM), которая применяется с помощью нагреваемого пластика, и selective laser sintering (SLS), которая использует порошкообразный материал для более сложных деталей.

3D-принтер – это станок, который управляется исполнительной программой и производит только аддитивные операции, добавляя порции материала к заготовке. В конечном итоге это позволяет получать трехмерные физические объекты.

Но, чтобы понять, как это реализуется под крышкой корпуса, обратимся к учебнику «Основы программирования станков с ЧПУ», которое поможет нам детально разобраться, какие знания требуются для полного понимания работы аппаратно-программного комплекса [4].

И так как 3D-печать попала в руки обычных пользователей, а не в промышленное производство, то существует множество вариаций прототипов.

В справочнике основ программирования станков с ЧПУ описывается, что такое G-code, благодаря чему агрегат или аппаратно-программный комплект выполняет порученные ему задачи [4].

G-code – это язык, который был создан более 50 лет назад и включен в стандарты, такие как ISO 6983–1:1982 и ГОСТ 20999-83. Этот язык активно используется производителями станков с ЧПУ, которые расширяют его по своему усмотрению, но основные коды и структура программы остаются неизменными. Программа, написанная на G-code, имеет жесткую структуру, в которой все команды

управления объединяются в кадры, состоящие из одной или более команд. Каждая команда включает в себя слова, объединяющая в себе латинские буквы и числовые значения.

Основные команды языка программирования начинаются с буквы G и включают действия, такие как позиционирование инструмента (G00 – G04), переключение рабочих плоскостей (G17 – G19), компенсация размера инструмента (G40 – G44), переключение систем координат (G53 – G59), циклы сверления и нарезания резьбы (G80 – G84) и переключение систем координат (абсолютная или относительная) с помощью команд G90 – G92[1].

Для принятия решения о выборе базы для разработки прототипа проведем анализ существующих 3D-комплексов для реализации физических 3D-моделей.

1.2 Анализ существующих аналогов 3D-комплексов

На сегодняшний день существует обширный выбор агрегатов для создания физических 3D-моделей [9].

Технология создания 3D-моделей находится в стадии развития и постоянного совершенствования. Широкий спектр возможностей, включая различные формы, размеры и установочное оборудование, делает её востребованной в различных отраслях. Многие промышленные сектора регулярно используют услуги 3D-печати, где в качестве материалов могут использоваться, например, бетонная смесь, металлический порошок и ABS-пластик.

Прототипирование на 3D-комплексе имеет множество преимуществ:

- высокая скорость создания прототипов;
- возможность скорость наращивания материала при помощи программной настройки;
- возможность использования различных материалов для печати;
- высокая точность прототипов 3D-моделей;
- возможность создания различных 3D-моделей;
- компактность аппаратно-программного 3D-комплекса;

- доступность как серийных комплексов, так и отдельных деталей;
- интуитивный интерфейс.

При создании прототипа агрегата с ЧПУ для печати 3D-деталей из ABS пластика были рассмотрены различные конфигурации 3D-принтеров, которые отличаются по своей кинематике движений. Самая распространенная система принадлежит классической модели Prusa Mendel, представленная на рисунке 5.

Рассмотрим одни из самых распространённых проектов и разберем их преимущества и недостатки.

1.2.1 Технические характеристики

Были рассмотрены такие технические характеристики 3D-комплексов таких как [9]:

- Flying Bear Ghost;
- Prusa Air;
- Ultimaker 2;
- Makerbot CupCake CNC;
- Rostock (Delta);
- Prusa Mendel.

Выбранная база прототипа должна обладать доступностью к изменению функциональных узлов на программном уровне. Чтобы аппаратная часть имела возможность к модернизации, агрегат должен обладать очень высокой жесткостью, возможностью к перепрошивке микроконтроллера и настройке каждого функционального узла под внесенные модернизации.

Параметры рассмотренных 3D-комплексов приведены в таблице 1.

Таблица 1 – Технические характеристики исследуемых 3D-комплексов

Система	Значения											
	Технология печати	Количество плавящийся головок	Область построения, мм	Скорость печати	Нагревательный стол	Интерфейсы	Дисплей	Материалы	Диаметр нити	Операционная система	Размер, мм	Вес, кг
Flying Bear Ghost	Моделирование методом наплавления (FDM/FFF)	1	230x230x210	150 мм/с	да	USB, SD-карта, WiFi	да	PLA пластик, HIPS Wood и другие.	1,75 мм	Windows, macOS, Linux	280x220x200	17
Prusa Air	Моделирование методом наплавления (FDM/FFF)	1	190x190x180	100 мм/с	да	SD-карта, USB	да	ABS и PLA пластик	1,75 мм	Windows, MacOS, Linux	430x405x 370	7,5
Ultimaker 2	Моделирование методом наплавления (FDM/FFF)	1	210x210x200	300 мм/с	да	Wi-Fi, USB, Ethernet	да	PLA пластик	1,75 мм	Windows, MacOS, Linux	338x358x389	16
Makerbot CupCake CNC	Моделирование методом наплавления (FDM/FFF)	1	252x199x150	130 мм/с	да	USB	да	ABS и PLA пластик	1,75 мм	Windows, MacOS, Linux	528x441x410	16
Rostock (Delta)	Моделирование методом наплавления (FDM/FFF)	1	300x300x600	300 мм/с	да	USB, SD-карта	да	ABS, PLA, PVA пластик, HIPS	1,75 мм	Windows, MacOS, Linux	430x550x1000	25
Prusa Mendel	Моделирование методом наплавления (FDM/FFF)	1	200x200x180	150 мм/с	да	USB тип B, SD-карта	да	ABS, PLA, HIPS, FilaFlex	1,75 мм	Windows, MacOS, Linux	460x383x430	9,7

1.2.2 Преимущества и недостатки 3D-комплексов

Таблица 2 – Плюсы и минусы аналогов 3D-комплексов

Система	Плюсы	Минусы
Flying Bear Ghost	<ul style="list-style-type: none"> – неплохая скорость печати (до 150 мм/с); – большая площадь печати (280x220x200 мм); – качественный экструдер, обеспечивающий стабильную работу и печать без засорений; – высокая степень автоматизации в работе, что делает принтер очень удобным в использовании. 	<ul style="list-style-type: none"> – частые сбой печатающей головки; – проблемы с фиксацией листа на столе при печати; – ограниченная поддержка материалов для печати; – малый выбор настроек, влияющих на качество печати.
Prusa Air	<ul style="list-style-type: none"> – высокая точность печати (до 50 микрон); – качественный привод в основе печати, обеспечивающий высокую скорость и качество печати; – алюминиевая рама, которая с легкостью стабилизирует колебания при печати; – отсутствие требований к характеристикам напряжения, тока и частоты сети. 	<ul style="list-style-type: none"> – нестабильность работы печатающей головки; – неполной комплектации и необходимость дозаказа комплектующих; – отсутствие изначальной поддержки системы охлаждения; – размер рабочей области.
Ultimaker 2	<ul style="list-style-type: none"> – высокое качество печати с высокой точностью и детализацией; – широкий выбор материалов для печати; – открытая система, позволяющая пользователю настраивать принтер на свое усмотрение; – имеет лазерный сенсор автоуровня; – скорость печати находится на высоком уровне. 	<ul style="list-style-type: none"> – высокая стоимость (это один из самых дорогих 3D-систем на рынке); – несмотря на высокое качество, печать занимает довольно много времени; – не поддерживает самые экзотические материалы; – высокий уровень шума.

Таблица 2 – Плюсы и минусы аналогов 3D-комплексов

Система	Плюсы	Минусы
Makerbot CupCake CNC	<ul style="list-style-type: none"> – достаточно компактный и легкий, что облегчает его транспортировку и работу в нескольких местах; – цена ниже, чем у большинства современных 3D-комплексов; – широкие возможности для модернизации и настройки; – позволяет использовать различные материалы для печати. 	<ul style="list-style-type: none"> – низкая точность и детализация печати по сравнению с более современными моделями; – ограниченный размер печатной платформы, что ограничивает размеры печатаемых объектов; – открытый возможный контакт с горячим экструдером; – нет автоматического уровня печати и калибровки.
Rostock (Delta)	<ul style="list-style-type: none"> – высокая скорость печати является одним из самых быстрых принтеров на рынке; – высокая точность и детализация печати; – сравнительно большая область печати, по сравнению с другими 3D-системами при одинаковом размере. – минимальные требования для калибровки стола перед началом печати; – низкий уровень шума, что делает его более пригодным для использования в домашних условиях. 	<ul style="list-style-type: none"> – высокая сложность монтажа и настройки, связанная с непривычной для многих конструкцией; – относительная негибкость в настройке и расширении, что делает его менее удобным для модернизации и улучшения; – трудности с калибровкой; – экструдер расположен на определенной высоте, что делает некоторые типы.
Prusa Mendel	<ul style="list-style-type: none"> – цена на данный принтер доступна широкому кругу пользователей, что делает его востребованным на рынке 3D-систем; – принтер построен на открытом исходном коде, что позволяет пользователям вносить изменения в программное обеспечение, настраивать параметры и оптимизировать работу устройства; – большое количество дополнительных опций и компонентов доступны для модификации принтера в соответствии с нуждами пользователя; – способен создавать детали и модели с высокой точностью и детализацией. 	<ul style="list-style-type: none"> – самостоятельная сборка может вызвать затруднения у пользователей, не имеющих опыта в сборке электроники; – максимальный размер моделей ограничен габаритами принтера; – поскольку Prusa Mendel является открытым проектом, необходимо проводить регулярную настройку принтера.

Из всех перечисленных качеств примеров аналогов 3D-комплексов, выделены следующие достоинства:

- цена на данный принтер доступна широкому кругу пользователей;
- принтер построен на открытом исходном коде, что позволяет пользователям вносить изменения в программное обеспечение, настраивать параметры и оптимизировать работу устройства.

В таблице 2 приведены основные достоинства и недостатки, на основе чего можно сделать выбор в пользу системы Prusa Mendel.

1.3 Анализ и выбор программно-аппаратного обеспечения системного назначения

Для того, чтобы аппаратно-программный комплект мог функционировать, а был не просто набором схем, проводов, и в целом, механизмов, на него надо загрузить программное обеспечение – образ, предназначенный для записи в память соответствующего устройства с целью обновления его встроенной программы, именуемый словом «прошивка».

Также, словом «прошивка» называют сам процесс записи этого образа в энергонезависимую память устройства. Такая прошивка памяти осуществляется при изготовлении устройства различными способами, например, установкой микросхемы памяти с предварительно записанным содержимым (прошивкой) либо внутрисхемным программированием микросхемы (например, микроконтроллера), уже установленной в устройство с помощью специального программатора.

На данный момент самыми распространенными прошивками являются «Marlin», «Teacup» и «Sprinter».

Существуют ответвления «Marlin», такие как «Marlin Kimbra» или «Marlin RichCattell», но они потеряли свою актуальность еще в начале 2015 года, тем самым рассматривать эти виды прошивок не имеет ни какого смысла.

Рассмотрим и проанализируем проекты с открытым кодом, варианты среды разработок и схем на микроконтроллерной основе в целях выявления оптимальных вариантов.

1.3.1 Открытый проект «Teacup»

Прошивка Teacup – это RepRap-прошивка с акцентом на производительность, стабильность, чистоту кода и простоту использования. Код написан на C++. Работает на большинстве RepRap-контроллеров на чипе ATmega и на некоторых контроллерах с ARM процессором [6].

Прошивка идёт с графическим интерфейсом Teacup Configtool который помогает выполнить конфигурацию, калибровку и загрузку прошивки (Рисунок 1).

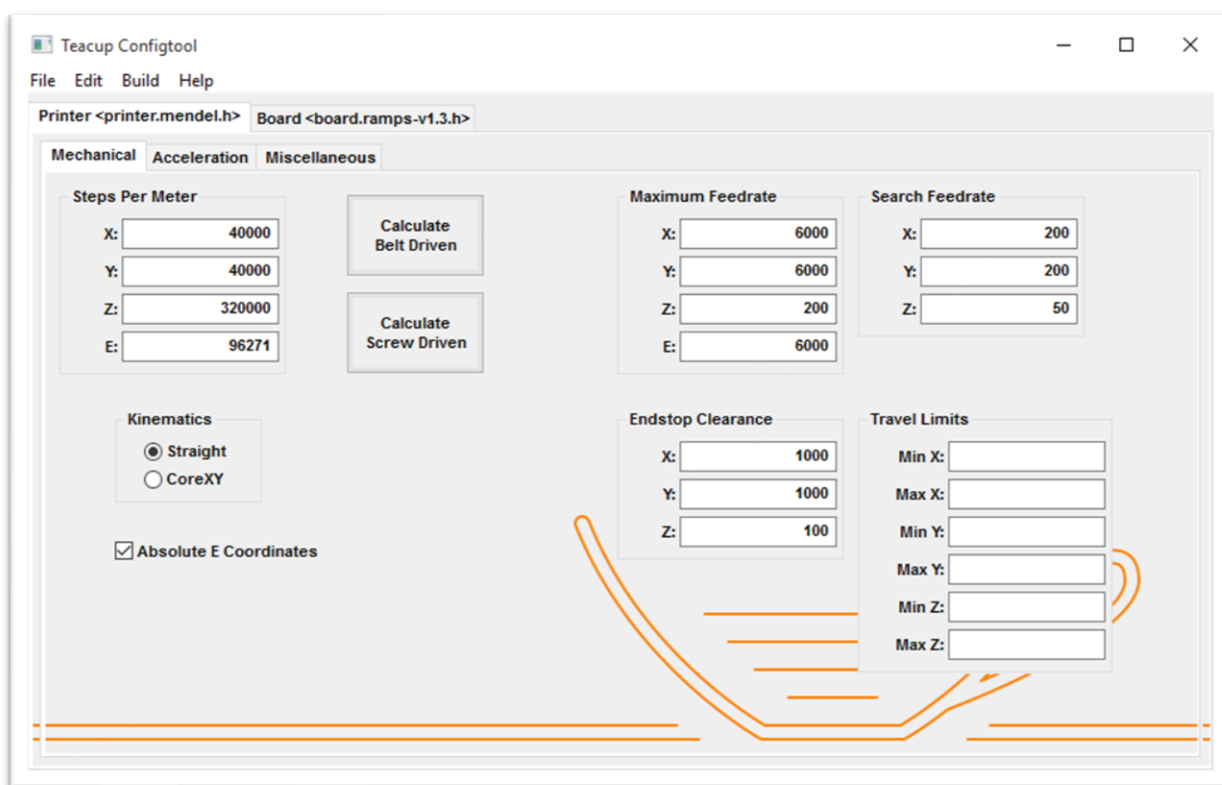
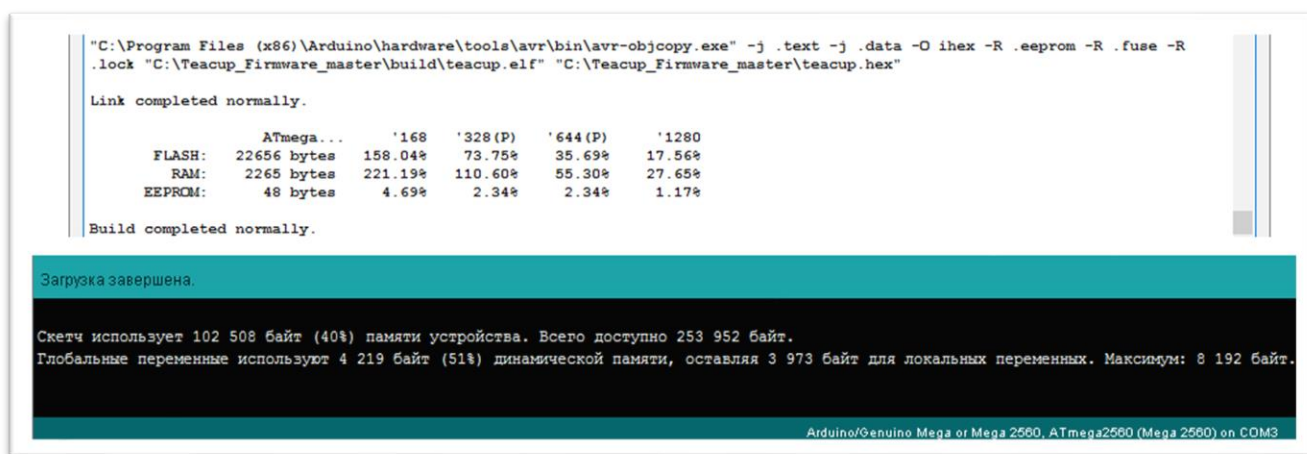


Рисунок 1 – Интерфейс Teacup Configtool

Лучше всего эта прошивка известна тем, кто пытался собрать 3D-комплекс на базе Arduino UNO, так как она оптимизирована на низкое потребление памяти (Рисунок 2).



```
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy.exe" -j .text -j .data -O ihex -R .eeprom -R .fuse -R .lock "C:\Teacup_Firmware_master\build\teacup.elf" "C:\Teacup_Firmware_master\teacup.hex"

Link completed normally.

      ATmega...    '168'  '328 (P)'  '644 (P)'  '1280'
FLASH: 22656 bytes 158.04% 73.75%  35.69%  17.56%
RAM:    2265 bytes 221.19% 110.60%  55.30%  27.65%
EEPROM: 48 bytes  4.69%   2.34%   2.34%   1.17%

Build completed normally.
```

Загрузка завершена.

Скетч использует 102 508 байт (40%) памяти устройства. Всего доступно 253 952 байт.
Глобальные переменные используют 4 219 байт (51%) динамической памяти, оставляя 3 973 байт для локальных переменных. Максимум: 8 192 байт.

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3

Рисунок 2 – Сравнение потребления памяти прошивок Teacup и Marlin

1.3.2 Открытый проект «Sprinter»

Прошивка Sprinter – это прошивка для контроллера 3D-принтера, который основан на платформе Arduino. Она представляет собой открытый исходный код (Open Source software), который позволяет пользователям самостоятельно настраивать параметры принтера [6].

Основная особенность прошивки Sprinter в том, что она предлагает точную калибровку 3D-принтера, а также позволяет реализовать более высокую скорость и точность печати. Это осуществляется путем настройки PID-регулятора, который контролирует температуру головки печати и стола.

В прошивке Sprinter также предусмотрена возможность автокалибровки стола, благодаря чему можно автоматически определить позиции X, Y, Z для наиболее точной печати.

Кроме того, Sprinter имеет простую и понятную структуру настроек, что делает использование ее более удобной. В прошивке Sprinter также поддерживаются

различные расширения и плагины. Возможность использования различных датчиков и энкодеров, позволяет настроить принтер под различные нужды и задачи.

Однако, если использовать прошивку Sprinter, надо брать в расчёт, что она не обновляется уже довольно длительное время, а именно с 2013 года, и может иметь некоторые недостатки, такие как: ограниченная поддержка функций и некоторые проблемы совместимости с новыми версиями Arduino IDE.

1.3.3 Среда разработки PlatformIO

PlatformIO – это среда разработки, которая позволяет разрабатывать и отлаживать код на разных платформах и микроконтроллерах, используя единый интерфейс и интегрированные инструменты [1].

С помощью PlatformIO можно разрабатывать программы на многих языках (C, C++, Python) для различных микроконтроллеров, таких как Arduino, ESP8266, ESP32, STM32 и многих других.

PlatformIO также предоставляет возможность тестирования и отладки кода на различных физических устройствах, что делает процесс разработки более удобным и эффективным.

Среда имеет интеграцию с популярными редакторами кода, такими как Visual Studio Code, Atom, Sublime Text и Eclipse, а также поддерживает автоматическую установку необходимых библиотек и зависимостей.

Плюсы PlatformIO:

- поддержка большого количества платформ и микроконтроллеров;
- интеграция с популярными редакторами: VSCode, Atom, Sublime Text и Eclipse;
- автоматическая установка необходимых зависимостей и библиотек;
- поддержка тестирования и отладки кода на физических устройствах;
- большое сообщество пользователей и разработчиков с открытым исходным кодом.

Минусы PlatformIO:

- не столь простой и понятный интерфейс, как у Arduino IDE;
- в некоторых случаях может быть более сложной в настройке, чем другие среды разработок;
- некоторые библиотеки и зависимости могут быть не совместимы или не работать должным образом в PlatformIO из-за специфики экосистемы;
- обновления могут приводить к конфликтам зависимостей и другим проблемам.

1.3.4 Среда разработки Arduino Eclipse plugin

Eclipse – это бесплатная среда разработки, которую можно настроить для работы с Arduino. Плагин для интегрированной среды разработки Eclipse, который облегчает процесс программирования приложений для платформы Arduino [1].

Плюсы:

- интеграция с мощной средой разработки Eclipse, что делает его более гибким и расширенным, чем Arduino IDE;
- поддержка нескольких языков программирования, таких как C, C++ и Python;
- широкий выбор платформ и микроконтроллеров для разработки;
- богатый набор инструментов для отладки, создания проектов и управления зависимостями;
- поддержка разных систем контроля версий (Git, SVN).

Минусы:

- его настройка может занять больше времени, чем простая установка Arduino IDE;
- ресурсоёмкость загрузки в оперативную память;
- интерфейс сложнее, чем Arduino IDE.

1.3.5 Платформа на базе Arduino Mega 2560

В качестве микроконтроллера мы используем ATmega2560 Arduino Mega [5]. Это является самой распространённой и легкодоступной платой, поэтому выбор пал на нее (Рисунок 3).

Arduino Mega построена на микроконтроллере ATmega2560 (техническое описание). Плата имеет 54 цифровых входов/выходов (14 из которых могут использоваться как выходы ШИМ), 16 аналоговых входов, 4 последовательных порта UART, кварцевый генератор 16 МГц, USB коннектор, разъем питания, разъем ICSP и кнопка перезагрузки.

Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, или подать питание при помощи адаптера AC/DC, или аккумуляторной батареи. Arduino Mega 2560 совместима со всеми платами расширения, разработанными для платформ Uno или Duemilanove (Таблица 3).

В целом, Arduino Mega 2560 – это мощный и гибкий микроконтроллер, который подходит для различных проектов, но может потребовать дополнительной работы для оптимизации памяти и стабильной работы с устройствами.

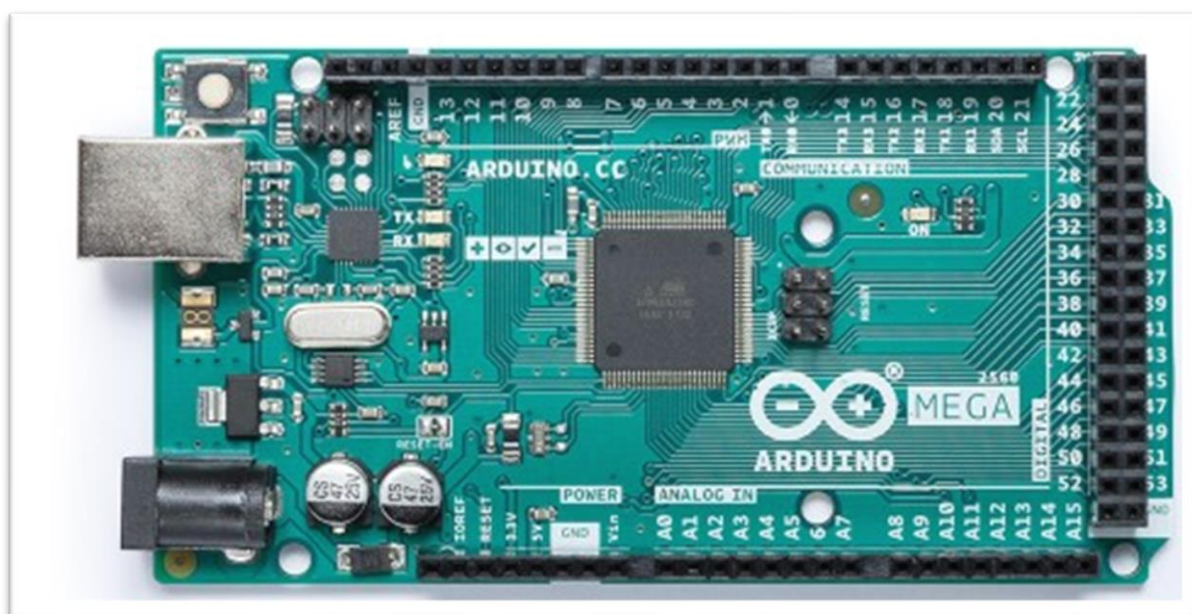


Рисунок 3 – Arduino Mega 2560

Таблица 3 – Технические характеристики платы Arduino Mega 2560

Наименование	Технические характеристики
Рабочее напряжение	5В
Входное напряжение (рекомендуемое)	7-12В
Входное напряжение (предельное)	6-20В
Цифровые входы/выходы	54 (14 из которых аналогичны ШИМ)
Аналоговые входы	16
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флеш-память	256 КВ (8 КВ для загрузки)
ОЗУ	8 КВ
Энергонезависимая память	4 КВ
Тактовая частота	16 МГц

Как итог, Arduino Mega 2560 идеально подходит для прототипирования 3D-комплекса, имеет большое количество функциональных улов, отлично совместима с открытыми разработками «Marlin» и ее стоимость колеблется от 2000 до 4000 рублей, что делает ее самым доступным и выгодным вариантом для начинающих и опытных разработчиков. Она обеспечивает широкие возможности для создания собственных устройств, в том числе систем реализации 3D-моделей, с использованием различных датчиков, модулей и других компонентов. Кроме того, Arduino Mega 2560 отлично подходит для обучения и изучения принципов работы программного обеспечения в компьютерной инженерии и электронике.

1.3.6 Плата расширения (Shield) Ramps 1.4

Ramps 1.4 – это электроника для управления 3D-комплексом, которая позволяет интерпретировать команды от контрольной платы и управлять моторами, нагревателями и другой периферией в 3D-комплексах. Ramps расшифровывается как "RepRap Arduino Mega Pololu Shield". Она установлена на Arduino Mega и использует драйверы шаговых моторов Pololu A4988 или DRV8825, которые обладают высокой производительностью и точностью шаговых движений (Рисунок 4).

Плата Ramps 1.4 имеет следующие возможности [5]:

– управление до 5 шаговыми моторами для позиционирования осей X, Y, Z и для экструдера;

– поддержка подключения до 2 нагревателей (для экструдера и платформы), датчика температуры для мониторинга температуры экструдера и платформы, концевых выключателей для концов осей X, Y, Z.

Ramps 1.4 является достаточно популярным и распространенным выбором для управления 3D-принтером, поскольку она подходит для большинства видов 3D-принтеров и существует множество подробных инструкций и схем подключения.

Одним из минусов может быть необходимость дополнительной настройки и калибровки для достижения наилучших результатов работы. Также использование более новых и продвинутых электронных плат, таких как SKR и MKS может быть более предпочтительным для более требовательных проектов.

И поскольку наша задача разработать аппаратно-программный комплекс, не используя дорогостоящие материалы, то выбор пал на самый бюджетный и довольно хорошем решении – связка Arduino Mega 2560 + Ramps 1.4.

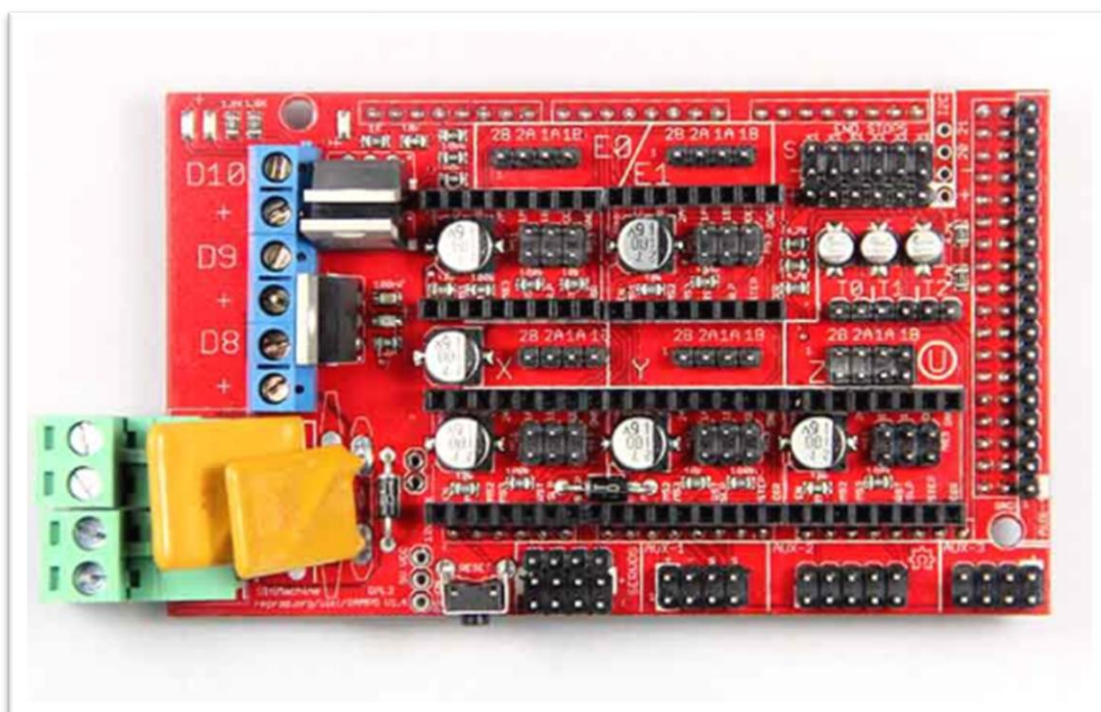


Рисунок 4 – Плата расширения Ramps 1.4

Еще одним преимуществом Ramps 1.4 является то, что эта электроника имеет открытый исходный код, которая позволяет оператором изменять и настраивать программное обеспечение в соответствии с их нуждами. Это дает большую свободу для творческих экспериментов и разработок.

Настройка Ramps 1.4 обычно достаточно проста, поскольку основные параметры могут быть настроены через файл конфигурации, который может быть изменен на персональном компьютере и загружен на контроллер печати по USB-порту.

Кроме того, Ramps 1.4 имеет дополнительные порты расширения, которые могут быть использованы для подключения дополнительных модулей и расширений, таких как датчики и экраны. В целом, это простой и надежный способ управления 3D-принтером для начинающих и опытных людей, которые работают с 3D-печатью.

Одним из недостатков Ramps 1.4 является ограниченный функционал по сравнению с более современными электронными платами, такими как SKR и MKS. Например, электроника Ramps 1.4 не поддерживает функцию автоматического нивелирования стола или детектора обрыва нити, необходимую для определенных типов 3D-печати.

Кроме того, электроника Ramps 1.4 имеет лимит по максимальному количеству моторов, которые можно подключить. В традиционной настольной печати, Ramps 1.4 может управлять только одним экструдером, что проблематично для тех, кто хочет печатать с разными материалами. Некоторые отдельные многодетальные печати с двумя экструдерами или более могут потребовать использования продвинутых электронных плат.

Наличие многочисленных проводов также может создать проблемы: каждый провод должен быть правильно подключен, и как правило, необходимы дополнительные усилия для экранирования, чтобы избежать помех и шумов в системе.

В целом, Ramps 1.4 является простым, недорогим и распространенным выбором для управления 3D-принтером, который может быть использован в качестве

отправной точки для более продвинутых и кастомизированных систем контроля печати. Если нужен такой функционал, можно рассмотреть более новые платы управления, такие как SKR или MKS.

1.4 Обоснование выбора принимаемого решения

На основе проведенного анализа, оценивания плюсов и минусов уже существующих аппаратно-программных комплексов, также открытых для общего доступа «прошивок» и сред разработок для адаптации открытого кода, были приняты следующие решения: «Prusa Mendel» как основа прототипа, среда разработки – Arduino IDE и в качестве базы для разработки ПО прототипа – открытый проект «Marlin».

1.4.1 Система Prusa Mendel

Из рассмотренных агрегатов и технических характеристик аналогов этих систем был выбран открытый проект Prusa Mendel (Рисунок 5).



Рисунок 5 – Система Prusa Mendel

Проект Prusa Mendel – это один из самых популярных открытых проектов 3D-принтера. Это проект, созданный главным конструктором компании Prusa Research – Йозефом Прусой. Prusa Mendel был разработан на основе оригинальной конструкции проекта RepRap и является его улучшенной версией [9].

Принтер Prusa Mendel отличается своей простотой и доступностью, он может быть собран и настроен даже начинающим пользователем. Большинство деталей для сборки можно напечатать на другом 3D-принтере, что делает его более доступным по стоимости, чем многие другие 3D-принтеры.

Проект Prusa Mendel постоянно улучшается и развивается, поэтому он до сих пор остаётся популярным в сообществе разработчиков 3D-принтеров. Йозеф Пруса перешел к разработке новых моделей принтеров, например, модели Prusa i3 и Prusa Mini. Проект Prusa Mendel был выпущен в открытом доступе в 2011 году и, с тех пор, он стал основой для создания многих других проектов.

На основе данной конструкции были созданы другие открытые проекты, такие как Printbot, Huxley и Ormerod.

Принцип работы принтера Prusa Mendel основан на технологии FDM (Fused Deposition Modeling), где термопластичный материал расплавляется и через сопло наносится на платформу. Принтер может использовать различные материалы – ABS, PLA, Nylon, TPU.

Существует множество различных модификаций Prusa Mendel, которые позволяют улучшить его производительность и качество печати. Например, модификация с обогреваемой платформой позволяет печатать материалы более высокой температуры, что повышает качество печати.

Несмотря на то, что проект Prusa Mendel уже не является самым новым и актуальным, он по-прежнему является одним из самых популярных проектов 3D-принтеров, благодаря своему открытому и доступному дизайну, а также своей простоте и надёжностью.

Данная система обладает компактностью, удобным расположением функциональных узлов. Также его цена не превышает 35 тысяч рублей, что говорит о его

доступности. Одним из наиболее важных критерий выбора является высокая скорость печатания, что отражено в его технических характеристиках (Таблица 4).

Prusa Mendel был создан более 10 лет назад, и его точность сильно уступает современным 3D-принтерам. Также Prusa Mendel имеет простую конструкцию, что делает его корпус менее устойчивым и могут возникать проблемы со стабильностью во время печати. Это может привести к искривлению печатных деталей и нарушению точности.

Основной компонент кинематики Prusa Mendel – это рама, состоящая из алюминиевых экструдированных профилей, образующих прямоугольную конструкцию, на которой расположены две основные оси – X и Y.

Также в кинематике используется ось Z, которая отвечает за перемещение нагревательного стола. Все три оси управляются шаговыми двигателями, которые перемещают насадки с помощью зубчатых ремней или шестеренок.

Таблица 4 – Технические характеристики Prusa Mendel

Технические характеристики	Значения
Размер рабочей зоны (X)	20 см
Размер рабочей зоны (Y)	20 см
Размер рабочей зоны (Z)	18 см
Используемый материал	PLA, HIPS, FilaFlex
Минимальная толщина слоя нити	60 мкм
Максимальная толщина слоя нити	300 мкм
Минимальная скорость печати	40-60 мм/с
Максимальная скорость печати	150 мм/с
Диаметр нити	1,75 мм
Интерфейсы	USB тип B, SD
Операционная система	Windows XP и старше, Mac OS X и старше, Linux
Программное обеспечение	Cura Software, Slic3r, Repetier, Kisslicer
Требования к электропитанию	220/12 В 100 В
Габаритные размеры принтера	460x383x430 мм
Удельный вес принтера	9,7 кг

Основной движущий механизм Prusa Mendel – это каретка, на которой расположен экструдер и головка печати. Каретка может двигаться вдоль оси Z, позволяя перемещать головку печати вверх и вниз (Рисунок 6).

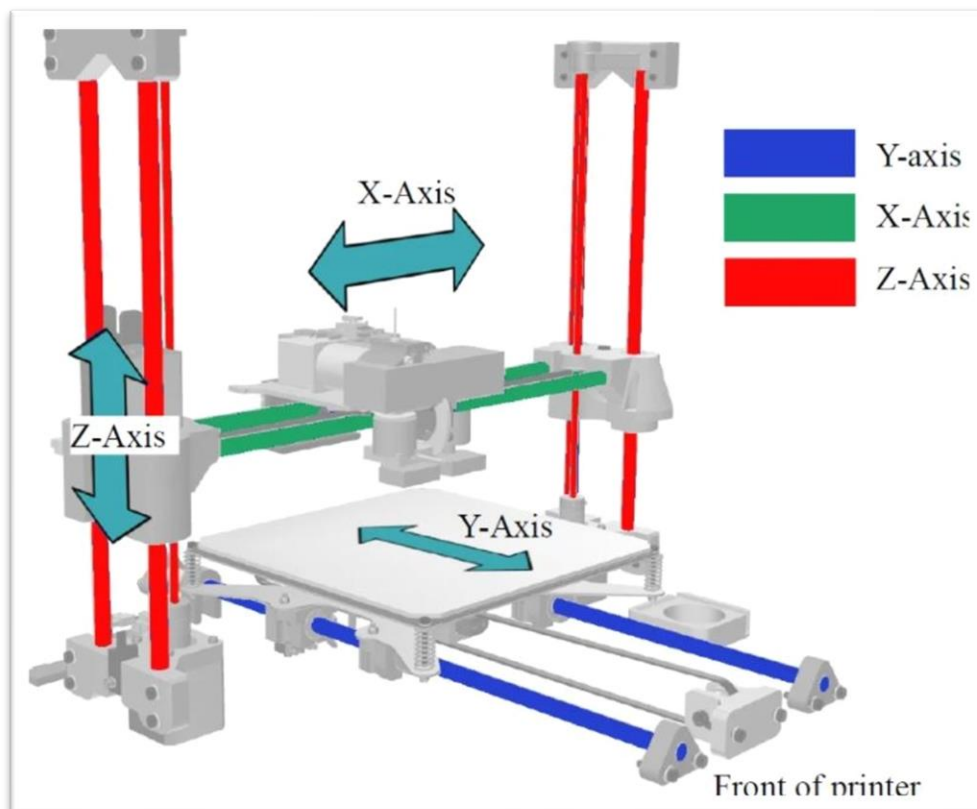


Рисунок 6 – 3D-модель прототипа электромеханического агрегата на базе Prusa Mendel

На оси X располагается подставка для печати, которая может двигаться вдоль оси Y, а на оси Y расположен каретный стол,двигающийся вдоль оси X.

Привод движения осей осуществляется шаговыми двигателями, которые устанавливаются в определенных точках рамы. Двигатели могут двигать оси с точностью до 0,1 мм. Для управления шаговыми двигателями используется электроника Arduino или RAMPS (плата расширения для Arduino), которые принимают сигналы от компьютера и драйверов шаговых двигателей.

Данная система не обладает хорошей термостабильностью, что может сказаться на качестве печати, особенно если в комнате сильно меняется температура.

Настройка и калибровка могут помочь улучшить качество печати, что делает данную модель наиболее подходящей для реализации.

В этой системе есть возможность модернизации и улучшения отдельных технических характеристик.

А благодаря Arduino и открытости программного кода это дает возможность вносить любые изменения, что позволяет дать волю фантазии и разработать прототип, отличающийся от всех других.

1.4.2 Среда разработки Arduino IDE

Arduino IDE – это программная среда разработки, использующая C++ и предназначенная для программирования всех плат ряда Arduino [3].

Аббревиатура IDE расшифровывается как Integrated Development Environment, в переводе – интегрированная среда разработки. С помощью этой среды программисты пишут программы, причем делают это гораздо быстрее и удобнее, чем при использовании обычных текстовых редакторов, хотя их тоже можно использовать для написания кода программ.

Arduino IDE позволяет составлять программы удобном текстовом редакторе, компилировать их в машинный код и загружать на все версии платы Arduino. Приложение является полностью бесплатным, а скачать его можно на официальном сайте сообщества Arduino.

Среда разработки Arduino IDE предоставляется бесплатно и поддерживается сообществом Arduino. Она совместима с большинством операционных систем, таких как Windows, Mac OS и Linux, что делает ее доступной для многих разработчиков. Эта среда разработки доступна всем, ее можно без каких-либо усилий скачать с официального магазина Microsoft.

Плюсы Arduino IDE:

- простой и понятный интерфейс, что делает его отличным выбором для начинающих разработчиков;
- интуитивный и понятный язык программирования на Arduino;
- широкое сообщество и большое количество библиотек и примеров;
- поддержка многих различных платформ, включая Arduino.

Минусы Arduino IDE:

- не такая гибкая, как другие среды разработки микроконтроллеров;
- ограниченные возможности отладки и тестирования кода;
- нет автоматической установки зависимостей и библиотек, что может быть неудобным для разработчиков со сложными проектами.

На основе анализа существующих сред разработок, Arduino IDE выделяется своей простотой, доступностью, и полностью совместима с платой Arduino Mega 2560 и ее расширением Ramps 1.4.

1.4.3 Открытый проект «Marlin»

Marlin – это прошивка с открытым исходным кодом, изначально разработанная для 3D-принтеров проекта RepRap FDM (моделирование плавным напылением) с использованием платформы Arduino [13].

Это, пожалуй, самая распространенная прошивка для 3D-принтеров. Она является открытой и бесплатной прошивкой, которая позволяет управлять 3D-принтером и оптимизировать его функциональность. Marlin обеспечивает широкий набор функций, включая поддержку различных типов 3D-принтеров, управление датчиками и возможность настройки печати. Прошивка также имеет популярные функции, такие как: автоматический уровень кровати (auto bed leveling), обнаружение обрыва нити (filament runout detection) и многие другие функции.

После того, как прошивка была скомпилирована из исходного кода C++, программа устанавливается и работает на материнской плате со встроенными компонентами и контактами ввода-вывода общего назначения для управления и связи с другими компонентами.

Для управления прошивка получает входные данные от USB-порта или подключенного носителя в виде команд G-кода, дающие машине понять, что делать. Например, команда G1 X10 указывает станку выполнить плавное линейное перемещение по оси X в положение 10. Основной цикл управляет всеми действиями станка в реальном времени, такими как: управление шаговыми двигателями через

драйверы шаговых двигателей, управление нагревателями, датчиками, индикаторами, дисплеем и пользовательским интерфейсом.

Marlin поддерживает множество различных типов платформ роботов для 3D-печати, в том числе базовые принтеры Cartesian, Core XY, Delta и SCARA, а также некоторые другие менее традиционные конструкции, такие как Hangprinter и Beltprinter.

Это, пожалуй, самая распространенная прошивка. Поддерживает множество кинематик, целый ряд дисплеев и разнообразную электронику, а также реализован автоуровень стола.

2 Описание объекта управления (электромеханического агрегата), разработка информационной модели процесса управления и аппаратно-программных моделей функциональных узлов разрабатываемой системы

2.1 Информационная модель процесса управления электромеханическим агрегатом

Вид процесса управления электромеханическим агрегатом включает в себя два информационных потока: информация о состоянии объекта, которая передается функциональными источниками, и информационный поток управления, который воздействует на функциональные исполнительные устройства, такие как шаговые двигатели, термисторы и концевые датчики для выполнения необходимых задач.

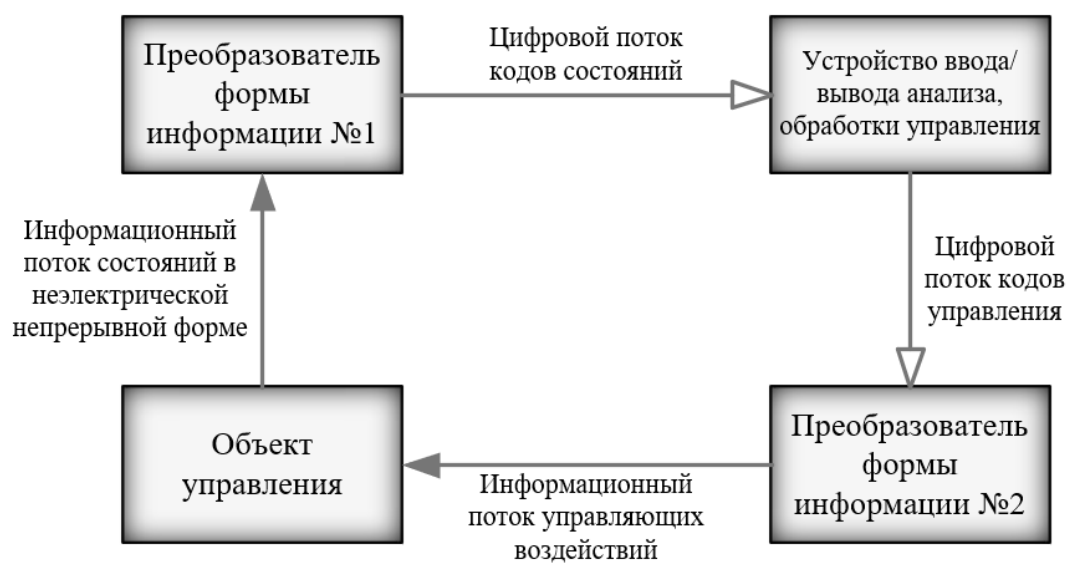


Рисунок 7 – Информационная модель процесса управления электромеханическим агрегатом

Структуру системы, обеспечивающую технологию реализации физических 3D-моделей, можно рассматривать как совокупность двух блоков (Рисунок 7):

- блок, обеспечивающий подготовку данных для работы системы программного управления комплексом;
- блок, отвечающий за объект управления, включающий микроконтроллер, который, в свою очередь, выполняет программные алгоритмы для шаговых двигателей, обеспечивающие перемещение экструдера и нагревательного стола по осям X, Y, Z.

Детализация блоков в структуре информационной модели процесса управления позволяет выделить обобщенную модель типового программно-функционального блока узла системы управления, представленная на рисунке 8.

Обобщенная модель описывает запрограммированный функциональный узел, который обращается к микроконтроллеру, а он в свою очередь, порождает управляющие воздействия на исполняющий функциональный узел (Рисунок 8):

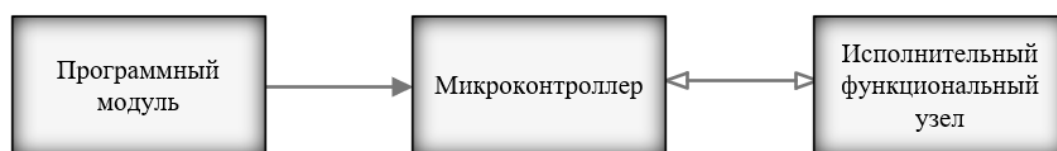


Рисунок 8 – Обобщенная модель взаимодействия программного модуля с микроконтроллером и исполнительным функциональным узлом

Программный модуль дает возможность инициации микроконтроллера и благодаря всем его средствам обеспечивает двунаправленное взаимодействие с исполнительным устройством.

Базируясь на информационной модели процесса управления, переходим на программный уровень и реализацию ПО, выделяя две задачи программирования:

- анализ цифрового потока кода состояний объекта управления;
- формирование управляющих воздействий на объект управления.

Процесс разработки ПО АПК на функциональном уровне представлен на рисунке 9.

Функциональные узлы, содержат регистровые структуры, позволяющие обеспечить базовые настройки.



Рисунок 9 – Процесс разработки программного обеспечения АПК на функциональном уровне

Выше приведена среда разработки, которая позволяет взаимодействовать с открытым кодом, в котором реализуется программирование функциональных узлов. После компиляции на выходе получаем программное обеспечение для прототипа 3D-комплекса.

2.2 Особенности программирования функционального узла «материнская плата»

При программировании функциональных узлов прототип необходимо проверить на правильность указанного идентификационного номера материнской платы в данной версии прошивки.

Нужно запрограммировать материнскую плату под конкретную конфигурацию. В собранной прошивке есть отдельный подготовленный раздел с тем, какой номер относится к определенной материнской плате. Для этого совершаем следующие действия:

- открываем раздел boards.h, и так как в данном случае используем «Ramps 1.4», то соответствующий номер это платы «43»;
- в соответствии с чем в дальнейшем меняем в нашем разделе конфигураций «configuration.h».

Список идентификационных номеров показан на рисунке 10.

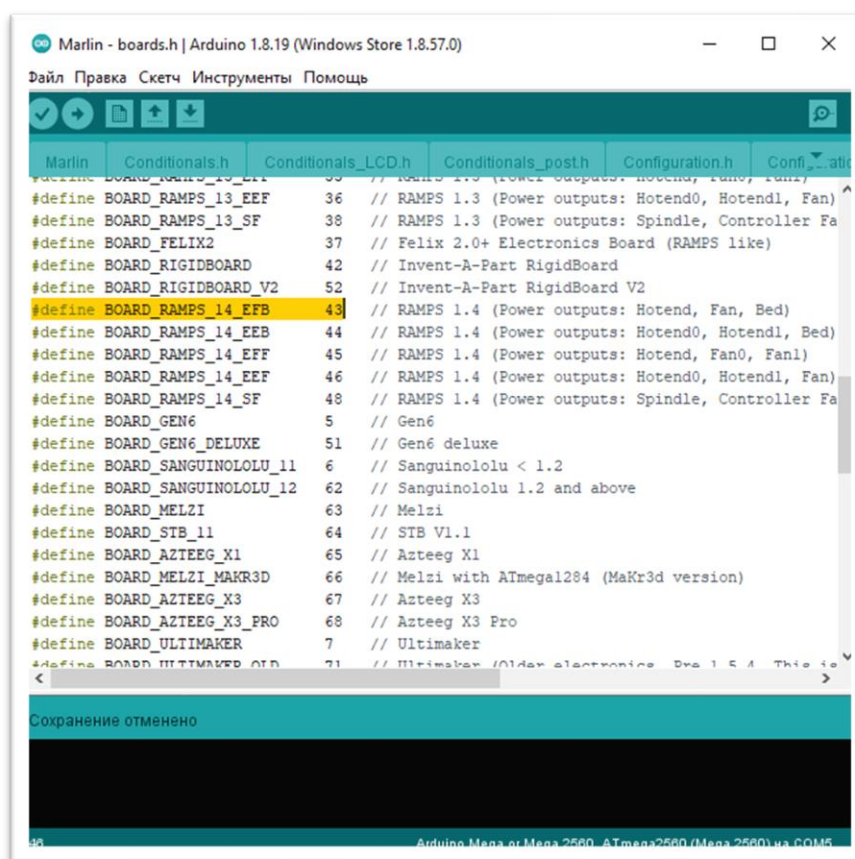


Рисунок 10 – Номер материнской платы

В соответствии с выбором аппаратной части, используем материнскую плату под идентификационным номером 43. Результат выбора в виде кода приведен ниже:

```
#ifndef MOTHERBOARD
```

```
#define MOTHERBOARD 43
#endif
```

Как видно из таблицы приведенной на рисунке 10, подключён: 1 экструдер, 1 вентилятор и нагревательный стол. 3 последних буквы обозначают какие элементы мы используем в нашей плате, например ЕЕВ – это экструдер + экструдер + нагревательный стол.

2.3 Программирование функционального узла «дисплейный модуль»

RepRapDiscount Smart Controller является популярным дисплейным модулем для 3D-комплексов, основанных на Arduino. Он имеет 128x64 пиксельный графический дисплей на жидких кристаллах (LCD), который обеспечивает удобный интерфейс, а также энкодер для управления 3D-комплексом.

С помощью Smart Controller можно изменять настройки принтера, смотреть информацию о прогрессе печати и управлять процессом печати. Также он имеет встроенный SD-кардридер, который позволяет загружать модели напрямую на 3D-комплекс без использования компьютера. Smart Controller является дополнительным модулем к Arduino Mega или RAMPS, совместимым с контроллером, установленным на штативе и позволяющим регулировать скорость, температуру и движение осей, а также запускать режим для предварительных тестовых действий. (Рисунок 11).



Рисунок 11 – Загрузочный экран

Чтобы Arduino могла в режиме реально времени показывать необходимые данные и управлять настройками с помощью дисплея, надо программно задать модель дисплея в части кода прошивки, проведенной ниже [11]:

```
#define REPRAP_DISCOUNT_SMART_CONTROLLER
```

Также для удобства интерфейс был переведен на русскую локализацию управления, но при загрузке прошивки дисплейный модуль не распознавал символы русского языка, поэтому пришлось возвращать на базовый английский язык.

Необходимо убедиться, что выбран русский язык в настройках.

Для этого нужно:

- зайти раздел «LCD and SD support»;
- выбрать "Русский" в качестве языка интерфейса;
- заменить части кода, а именно «en» на «ru».

Часть кода отвечающая за смену языка интерфейса приведена ниже:

```
#define LCD_LANGUAGE en
```

Для того, чтобы было удобно управлять энкодером необходимо, чтобы при повороте его влево, индикатор дисплейного модуля следовал по меню вниз, а при повороте вправо – он перемещался вверх.

В меню мониторинга, с целью удобства регулирования температуры, после вращения энкодера по часовой стрелке, осуществлялось увеличение значения показателей, против часовой стрелки – уменьшение, а также тестовое вращение шаговыми двигателями. Часть кода, отвечающая за инвертирование энкодера, приведена ниже:

```
#define REVERSE_ENCODER_DIRECTION  
#define REVERSE_MENU_DIRECTION
```

Звуковой канал используется при окончании печати и при вращении энкодера. Программное указывается в коде, приведенном ниже:

```
#define SPEAKER
```

2.4 Пример программного подключения функционального узла «термистор»

В разделе «Thermal Settings» вписываются термисторы, которые были установлены. Нас интересуют строки, относящиеся к нашему экструдеру и столу, тем самым, задавая параметры в прошивке на определенные термисторы.

Важно отметить, что использование качественного термистора HotEnd и правильная его калибровка, оказывают влияние на качество и точность печати.

Поэтому при замене термистора HotEnd необходимо выбрать аналогичный по техническим характеристикам и не забывать о его калибровке. Результат выбора программного значения термистора показан на рисунке 12.

Пример параметра термистора HotEnd или экструдера [11], показан ниже:

```
#define TEMP_SENSOR_0 5
```

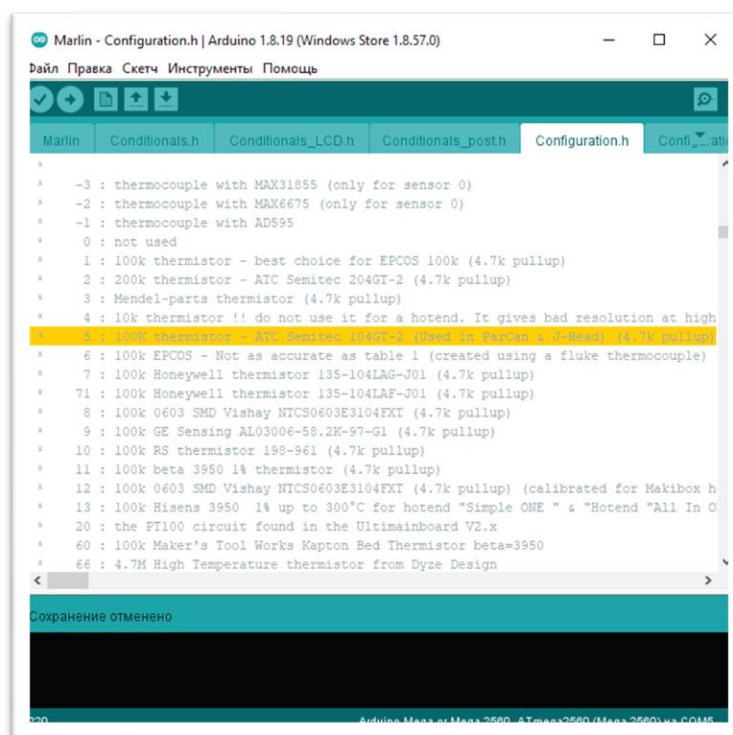


Рисунок 12 – Выбор значения термисторов

Пример параметра термистора стола [11], приведен ниже:

```
#define TEMP_SENSOR_BED 11
```

Так как мы используем один экструдер и стол, то нам и нужно выставить всего 2 термистора и указать их в коде для контроля и мониторинга нагрева.

После выставляем максимальную и минимальную температуру экструдера и стола. Детальный разбор конфигураций стола и экструдера указан в разделе 3.5 и 3.6.

2.5 Программные ограничения функционального узла «экструдер»

Температура экструдера (HotEnd) является одним из наиболее важных параметров для качественной печати на 3D-принтере. Она определяет, насколько горячим будет пластик при выходе из сопла и как он будет распределяться по столу.

Максимальная температура экструдера на принтере Prusa Mendel обычно составляет 260-270°C. Однако, рекомендуется не превышать 250°C, так как это позволяет избежать проблем с накалом нагревательного блока и гниением материала.

Минимальная температура экструдера на принтере Prusa Mendel зависит от материала, который вы собираетесь использовать. Например, для PLA обычно достаточно температуры в диапазоне 185-210°C, а для ABS – 225-240°C. Поэтому необходимо установить температуру экструдера, соответствующую требованиям материала, которую хотите использовать, что обеспечит правильное экструдирование.

При выборе температуры экструдера следует учитывать, что слишком низкая температура может привести к затвердеванию материала в сопле экструдера, а слишком высокая температура может вызвать застывание материала и засорение сопла экструдера.

При программировании необходимо обратить особое внимание на температуру экструдера и не допускать перегрева, так как это может привести к повреждению нагревательного блока и других частей принтера.

Важно отметить, что конкретные значения минимальной и максимальной температуры экструдера могут немного отличаться.

Именно поэтому выставляем значения в соответствии с температурными ограничениями модели экструдера.

Минимальная температура экструдера:

```
#define HEATER_0_MINTEMP 5
#define HEATER_1_MINTEMP 0
#define HEATER_2_MINTEMP 0
#define HEATER_3_MINTEMP 0
```

Максимальная температура экструдера:

```
#define HEATER_0_MAXTEMP 260
#define HEATER_1_MAXTEMP 0
#define HEATER_2_MAXTEMP 0
#define HEATER_3_MAXTEMP 0
```

Но экструдер не начнет работу, пока он не нагреется до определенной температуры, и там самым запустится процесс печати. Следовательно, в коде выставляем минимальную температуру, при которой разрешено использование экструдера [11]:

```
#define PREVENT_COLD_EXTRUSION
#define EXTRUDE_MINTEMP 170
```

Изначально его надо откалибровать, чтобы подача филамента была правильной. Выставляем значение на «5», а уже после калибровки возвращаем на оптимальное значение в «170» [11]:

```
#define EXTRUDE_MINTEMP 5
```

2.6 Программирование функционального узла «нагревательный стол»

Минимальная температура нагревательного стола 3D-принтера Prusa Mendel – 0°C, а максимальная температура – 110°C. Однако, рекомендуется не превышать температуру 90°C, чтобы избежать деформации и дефектов печати.

При работе с нагревательным столом необходимо учитывать материал, который используется при печати, а также тип поверхности стола. Например, для печати ABS-пластика необходима повышенная температура стола, чем для печати PLA-пластика.

Ниже приведены программируемые параметры:

- диапазон температур преднагрева;
- диапазон температур для ABS-пластика;
- диапазон температур для PLA-пластика;
- интенсивность системы охлаждения.

Таким образом, минимальную температуру нагревательного стола выставляем на 5°C [11]:

```
#define BED_MINTEMP 5
```

А максимальную на 110°C:

```
#define BED_MAXTEMP 110
```

Еще одним важным пунктом является настройка преднагрева. Для PLA-пластика выставляем одни параметры температуры, а для ABS-пластика – повышенные значения.

Первые настройки для PLA-пластика, тут выставляем температуру HotEnd 200°C [11]:

```
#define PREHEAT_1_TEMP_HOTEND 200
```

Температуру стола изменяем на 70°C:

```
#define PREHEAT_1_TEMP_BED 60
#define PREHEAT_1_FAN_SPEED 0 // Value from 0 to 255
```

Следом идут настройки для ABS-пластика, тут температуру HotEnd повышаем до 250°C, а температуру стола до 110°C:

```
#define PREHEAT_2_TEMP_HOTEND 240
#define PREHEAT_2_TEMP_BED 100
#define PREHEAT_2_FAN_SPEED 0 // Value from 0 to 255
```

Скорость вентиляторов оставляем по нулям, так как обдув будет включаться непосредственно во время печати.

2.7 Программирование параметров функционального узла «концевой переключатель»

На основе выбранных концевых переключателей, которые нормально замкнуты, выставляем параметры «true» в части кода прошивки, приведенной ниже [11]:

```
#define X_MIN_ENDSTOP_INVERTING true
#define Y_MIN_ENDSTOP_INVERTING true
#define Z_MIN_ENDSTOP_INVERTING true
#define X_MAX_ENDSTOP_INVERTING true
#define Y_MAX_ENDSTOP_INVERTING true
#define Z_MAX_ENDSTOP_INVERTING true
```

Для концевых переключателей требуется настройка расположения.

Для этого переходим в раздел «ENDSTOP SETTINGS». Все концевые переключатели находятся в минимуме – в нулевой позиции координат, поэтому менять ничего не требуется. Но в случае необходимости вместо перемещения концевого

переключателя в противоположную сторону, легко программно сменить его расположение относительно нулевой координаты [11]:

```
// ENDSTOP SETTINGS:  
// Sets direction of endstops when homing; 1=MAX, - 1=MIN  
// :[- 1, 1]  
#define X_HOME_DIR - 1  
#define Y_HOME_DIR - 1  
#define Z_HOME_DIR - 1
```

2.8 Особенности программирования системы движения осей

Система электромеханического агрегата, отвечающая за определение ранее заданного пути движения стола и печатающей головки для создания трехмерной модели.

Для этого некоторые 3D-комплексы могут использовать комбинированные кинематические системы, например, одновременно применять прямолинейно-зубчатую и Delta-кинематики. Такие системы позволяют достичь высокой точности печати и помогают реализовать более сложные геометрические формы в печати.

У разных моделей систем кинематика разная и в прошивке прописано 6 видов, но не один из пунктов кинематик, прописанный в коде не требуется, так как каждая ось у нас независима друг от друга, то добавляется седьмая «независимая», что, в свою очередь, является особенностью разработанного прототипа. Поэтому код прошивки изменяем следующий образом: удаляем или комментируем данный блок кода прошивки, приведенный ниже [11]:

```
//#define COREXY  
//#define COREXZ  
//#define COREYZ  
//#define COREYX  
//#define COREZX  
//#define COREZY
```

2.8.1 Математическое обеспечение механизма движения осей

Далее, приступаем к настройке движения осей. Тут необходимо указать количество шагов для сдвига на миллиметр.

Сдвиг осей X и Y на 1 миллиметр просчитывается делением полного оборота вала двигателя на расстояние, пройденное за полный оборот:

$$S_{\text{на миллиметр}} = \frac{P * J}{S_{\text{полное}}}, \quad (1)$$

где P – количество шагов полного оборота;

J – деление шага;

S – пройденное расстояние.

Используемые моторы имеют 200 шагов на оборот, также у нас включено деление шага на 16, отсюда следует: чтобы получить количество шагов на полный оборот, нужно 200 умножить на деление шага, а потом разделить на пройденное расстояние. Здесь используется ремень «GT2» (Рисунок 13), у которого шаг зубьев 2 миллиметра (Таблица 5), а на шкиве, в свою очередь, 20 зубьев. Поэтому $S_{\text{полное}}$ считается следующим образом:

$$S_{\text{полное}} = P_3 * K_{\text{ш}}, \quad (2)$$

где P_3 – шаг зубьев;

$K_{\text{ш}}$ – количество зубьев на шкиве.

Таблица 5 – Характеристики ремня GT2

Наименование	Технические характеристики
Ширина ремня	6мм
Толщина	1.38мм
Шаг	2



Рисунок 13 – Ремень GT2

Для получения расстояния, пройденного за 1 оборот вала, перемножаем эти значения:

$$S_{x,y} = \frac{200 * 16}{2 * 20} = 80.$$

Исходя из этих значений, получаем 80 и вписываем это значение для осей X и Y, что обозначает 80 шагов на движение на 1 миллиметр [11]:

```
/**  
 * Default Axis Steps Per Unit (steps/mm)  
 * Override with M92  
 *           X, Y, Z, E0 [, E1[, E2[, E3]]]  
 */  
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, ____, ____ }
```

Но для оси Z необходимо количество шагов полного оборота поделить на шаг винта. Количество шагов для полного оборота также, как и в предыдущем случае,

только деление шага на 4. Шаг винта 1 миллиметр, так как мы используем винт М6 (для резьбовой шпильки М8 шаг будет 1.25).

$$S_z = \frac{200 * 4}{1} = 800.$$

И меняем значение 400 на 800:

```
/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *           X, Y, Z, E0 [, E1[, E2[, E3]]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 800, ____ }
```

Далее идёт экструдер. Рассчитываем количество шагов для подачи филамента на 1 миллиметр. Для этого количество шагов полного оборота подающей шестерни делим на длину окружности подающей шестерни. Количество шагов для оборота подающей шестерни считается по формуле, где количество шагов полного оборота вала двигателя необходимо умножить на деление шага и на редукцию.

Редукция – это отношение количество зубов ведомой шестерни к количеству ведущей (Рисунок 14).



Рисунок 14 – Ведущая и ведомая шестерни экструдера

Получаем следующую формулу:

$$S_3 = \frac{P_B * J * \Phi}{\pi * D}, \quad (3)$$

где S_3 – расстояние пройденного оборота для подачи 1мм филамента;

J – деление шага;

Φ – редукция, которая, в свою очередь, считается по формуле (4);

π – постоянная, равная 3,14;

D – диаметр вала.

В нашем случае длина окружности вала равна 33.3 (Рисунок 15).



Рисунок 15 – Длина окружности вала

$$\Phi = \frac{K_{\text{ведомой}}}{K_{\text{ведущей}}}, \quad (4)$$

где $K_{\text{ведомой}}$ – количество зубьев ведомой шестерни;

$K_{\text{ведущей}}$ – количество зубьев ведущей шестерни.

Но, так как в нашем случае отсутствует ведомая шестерня, редукция у нас равна 1, и в итоге получаем:

$$S_3 = \frac{200 * 16 * 1}{33.3} = 96.$$

96 шагов требуется для проталкивания 1мм филамента. Полученный результат вписываем в код прошивки [11]:

```
/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *           X, Y, Z, E0 [, E1[, E2[, E3]]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 800, 96 }
```

2.8.2 Программная настройка скорости подачи и ускорения

Настройка скорости подачи и ускорения является важным шагом в настройке 3D-принтера. Программируемые параметры определяют, насколько быстро движется экструдер по осям X, Y и Z, а также скорость подачи пластика. Эти параметры могут существенно повлиять на качество печати, скорость и стабильность движения экструдера.

После, выставляем максимальную скорость подачи [11]:

```
#define DEFAULT_MAX_FEEDRATE { 300, 300, 3, 300 }
```

А также максимальное ускорение и значение ускорения по умолчанию [11]:

```
#define DEFAULT_MAX_ACCELERATION { 1000, 1000, 100, 10000 }
#define DEFAULT_ACCELERATION      1500
#define DEFAULT_RETRACT_ACCELERATION 3000
#define DEFAULT_TRAVEL_ACCELERATION 3000
```

2.8.3 Программное управление инвертированием движения шаговых двигателей

Выбранный открытый проект прошивки позволяет программно инвертировать движение моторов по осям. Таким образом, движение по оси идёт в плюс, а надо наоборот. А так как переставлять шаговый двигатель не имеет возможности или не имеет смысла, то можно просто заменить в коде «true» на «false» в зависимости от того, в какую сторону должна двигаться ось. Поскольку расположения двигателей выбраны так, что все оси двигаются в одном направлении, мы выставим инвертирование на «true» [11]. Элемент кода показан ниже:

```
#define INVERT_X_DIR true
#define INVERT_Y_DIR true
#define INVERT_Z_DIR true
```

В этом же разделе можно инвертировать и движение мотора экструдера. Но, так как изначально установленный мотор двигался не в том направлении, пришлось его инвертировать, отвечающая за инвертирование часть кода привереда ниже [11]:

```
#define INVERT_E0_DIR false
```

2.8.4 Программное управление областью печати

Область печати 3D-принтера определяет максимальный размер объекта, который может быть напечатан на данном принтере. Эта область обычно определяется как объем, который может заполнить печатающая головка внутри

определенных границ. Область печати может изменяться в зависимости от модели принтера и производителя. Размеры могут быть выражены в миллиметрах, сантиметрах или дюймах, и могут варьироваться от небольших размеров для домашних принтеров до более крупных размеров для промышленных принтеров. Область печати 3D-принтера является важным фактором при выборе принтера и определении возможностей для напечатанных объектов.

Так как наш прототип разработан на основе открытого проекта Prusa Mendel, область печати достаточно скромная, и чтобы печать не выходила за рамки нагревательного стола, следует программно прописать область печати.

Для прототипа выбран алюминиевый нагревательный стол МК3 (Рисунок 16).



Рисунок 16 – Нагревательный стол МК3

- размеры: 214x214 мм;
- толщина: 3 мм;
- напряжение питания: 12 В.

Так как у нас концевые переключатели стоят в начале координат осей, не давая выйти за пределы области печати, задаем габариты по всем осям:

- ось «X» равна 200мм;
- ось «Y» равна 200мм;
- ось «Z» равна 180мм.

Чтобы не допустить выхода за пределы области печати и обеспечить правильное размещение объекта на столе принтера, необходимо задать размеры габаритов по всем осям (X, Y, Z). Это можно сделать в программе для подготовки модели (например, в Cura), а именно в коде прошивки для разработанного прототипа.

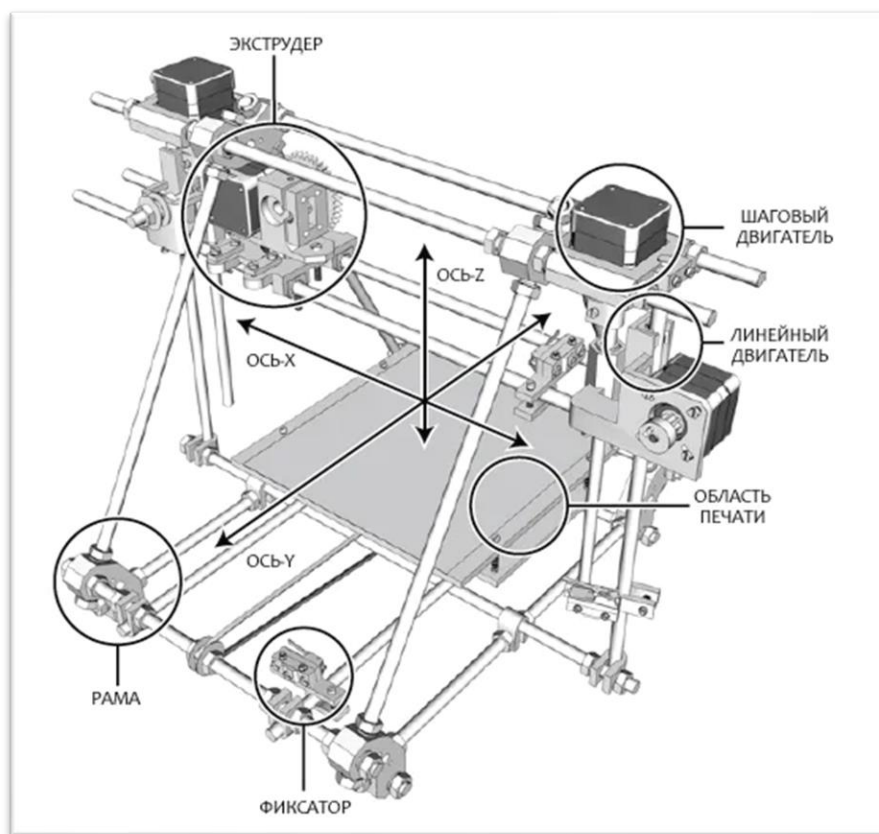


Рисунок 17 – 3D-модель области печати

В коде указываем границы в миллиметрах, тем самым плавающая головка экструдера не выйдет за границы [11]. Часть кода, отвечающая за пределы области печати, приведена ниже:

```
#define X_MIN_POS 0
#define Y_MIN_POS 0
#define Z_MIN_POS 0
#define X_MAX_POS 200
#define Y_MAX_POS 200
#define Z_MAX_POS 180
```

Получается, что область печати составляет 0,0072 м³.

2.9 Программное подключение функции распознавания SD-карт

Для работы с SD-картой, необходимо в коде прошивки, в блоке "LCD and SD support" включить поддержку чтения с SD-карты. Для этого нужно установить значение "true" в параметре "SD_SUPPORT", а также задать пины для подключения SD-карты к принтеру. Обычно это делается через интерфейс SPI, поэтому необходимо указать номера пинов для подключения "MOSI", "MISO", "SCK" и "SS". Кроме того, можно настроить скорость передачи данных с карты, указав нужное значение в параметре "SD_SPEED"[11]. Элемент кода показан ниже:

```
#define SDSUPPORT
```

После проверяем, распознается ли SD-карта, и запускаем с нее, заранее загруженный проект для печати (Рисунок 18).

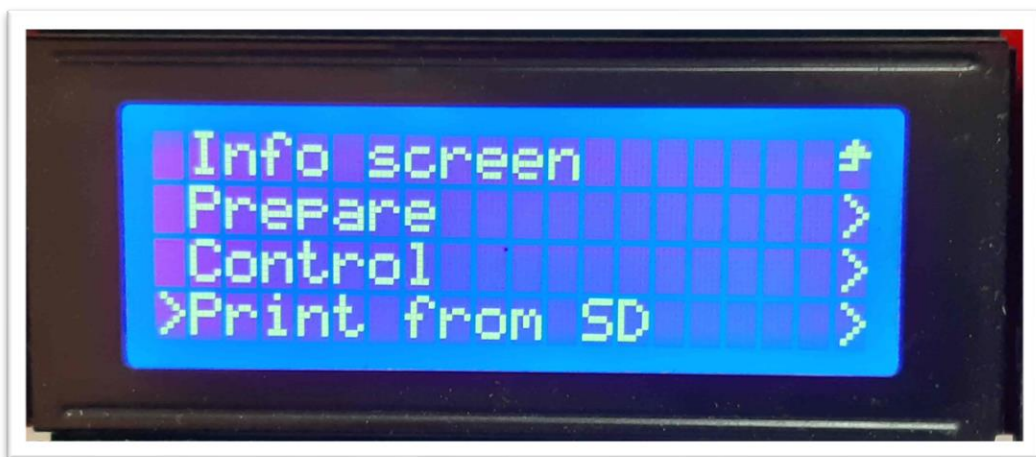


Рисунок 18 – Распознавание SD-карты

Если всё сделано верно, то начнётся процесс подготовки к печати, а далее – сама печать. Не мало важный элемент проверки – читается ли SD-карта. Это необходимо для того, чтобы используемый G-code отработал алгоритм от начала до конца и для этого лучше дождаться окончания печати.

3 Тестирование программного обеспечения для разрабатываемого комплекса

Для проверки программного обеспечения были использованы программы калибровки и тестирования, такие как:

- Pronterface;
- Fusion 360;
- UltiMaker Cura;

Подробные процедуры программной калибровки 3D-комплекса помогут достичь наилучших результатов печати. Они позволяют управлять всеми аспектами процесса печати, обеспечивая максимальную точность, скорость и качество изделий.

Процедура калибровки осуществляется при помощи специализированной программы – Pronterface [12].

Pronterface – это программное обеспечение управления 3D-принтером, которое позволяет пользователям отправлять команды на принтер, управлять параметрами печати и просматривать текущий статус принтера. Оно является одним из самых популярных программных решений для управления 3D-принтерами и поддерживается большинством принтеров на рынке. Pronterface предоставляет пользователю интерфейс для управления принтером через соединение USB, а также может быть использован для запуска печатных операций с SD-карты.

Перед началом калибровки обязательно необходимо проверить движение осей, используя Pronterface (Рисунок 19).

После запуска программы Pronterface, выбираем порт и скорость передачи данных и нажимаем подключиться. Пример использования приведен на последующих рисунках.

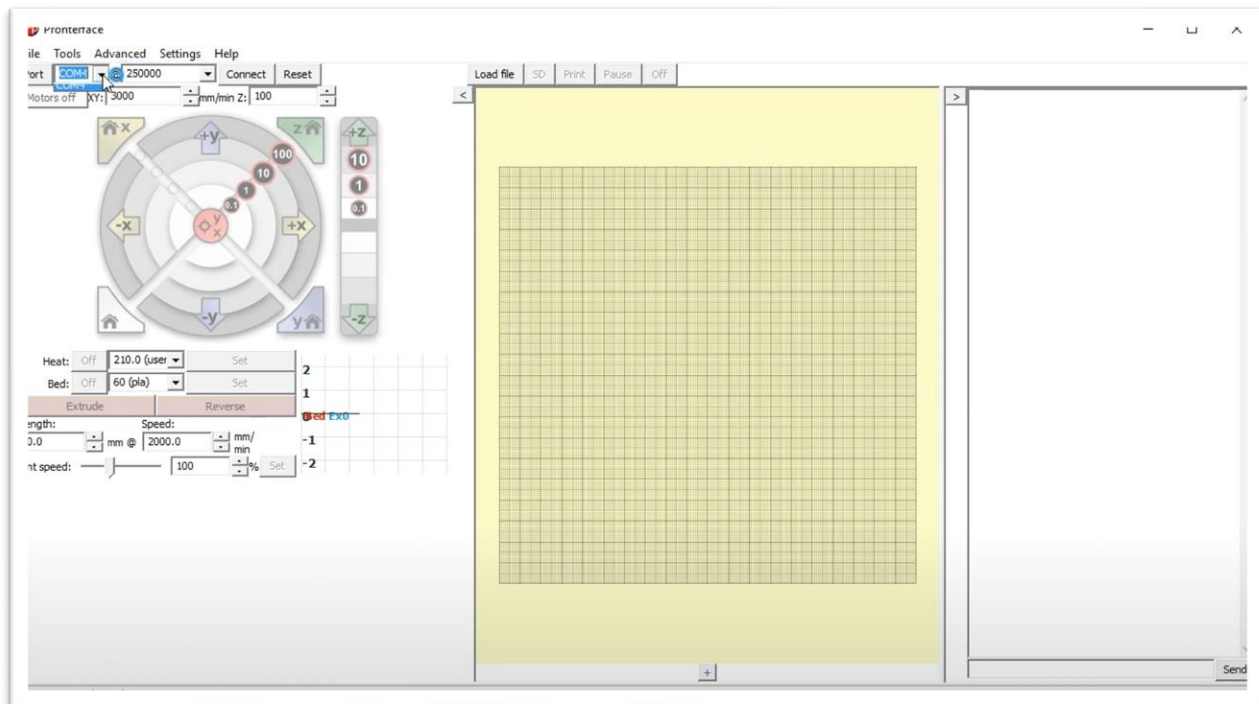


Рисунок 19 – Выбор порта в графической среде Pronterface

Далее, двигаем оси и проверяем, в правильном ли направлении они двигаются (Рисунок 20).

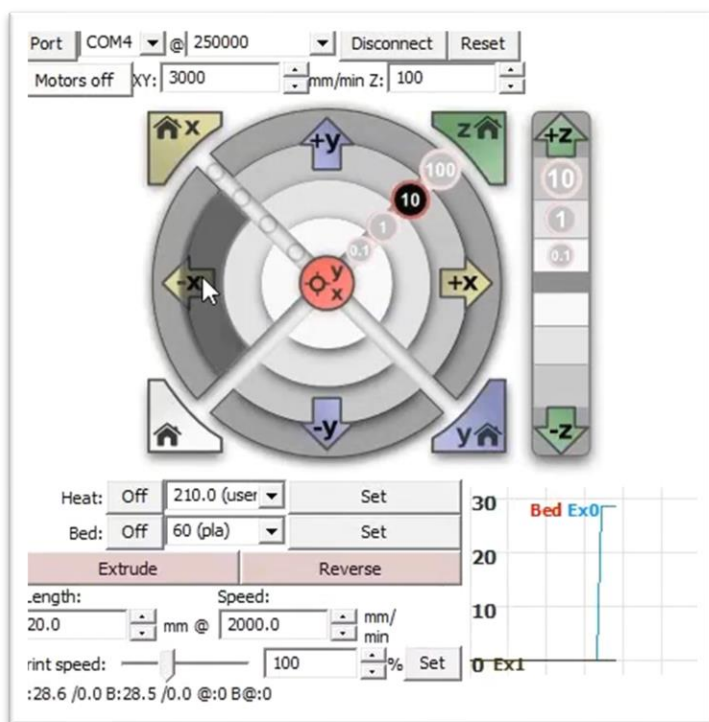


Рисунок 20 – Программное управление осями

3.1.1 Программная калибровка PID-контроллера температур

Чтобы откалибровать PID-контроллер температур, возвращаемся в прошивку. В разделе «PID Settings» и раздел «PID > Bed Temperature Control».

Тут есть нужная команда «M303 E-1 C8 S90»:

где M303 – команда автокалибровки;

E-1 – нагреваемый стол (в свою очередь, E-0 будет означать – экструдер);

C8 – количество циклов;

S90 – температура калибровки.

Далее, вставляем данную команду в Pronterface.

Начнем мы с калибровки температуры экструдера. Для этого меняем значения команды на следующие: M303 E0 C8 S260. Замена значений означает то, что мы выбрали процесс калибровки экструдера и его температуру в 260°C.

Нажимаем «отправить», или в данном случае – это кнопка «send» и видим, как на графике температура начала расти (Рисунок 21).

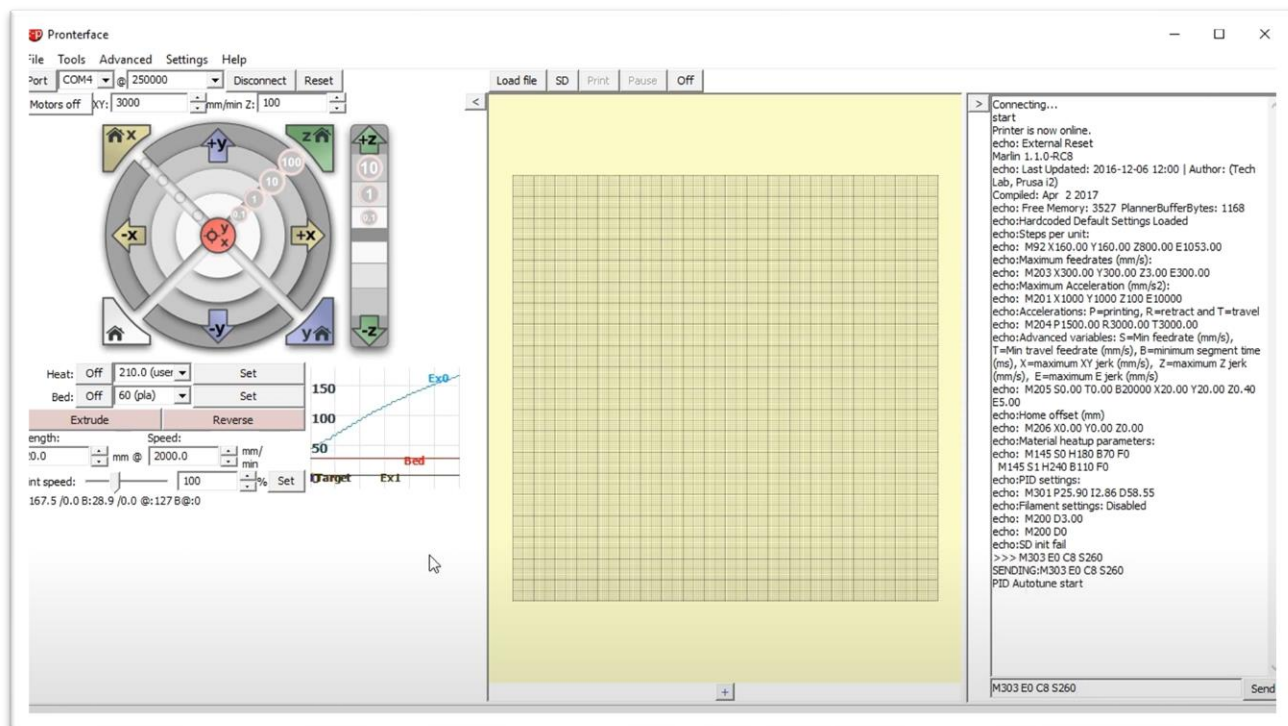


Рисунок 21 – Программное управление калибровкой

Дожидаемся результатов автокалибровки и в конце получаем результаты (Рисунок 22).

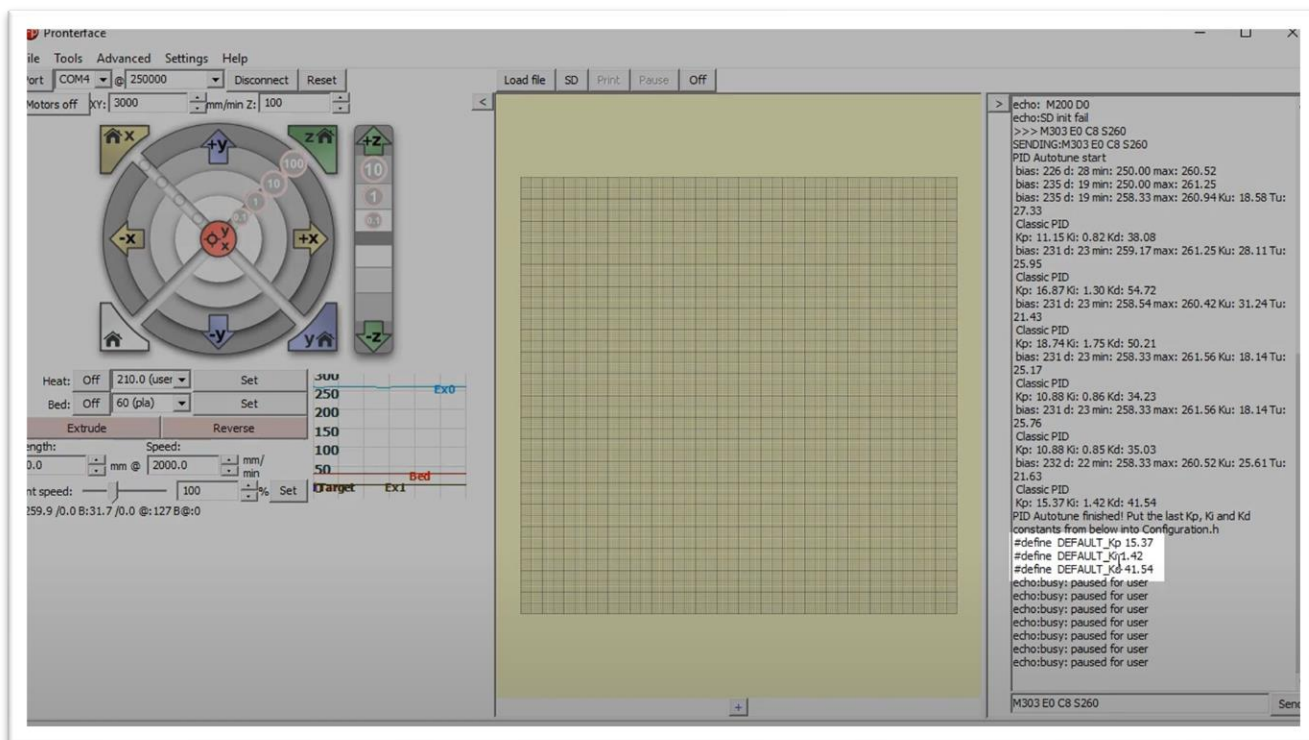


Рисунок 22 – Результат автокалибровки

Получение результаты переносим в код прошивки. Часть кода показана ниже:

```
// Prusa
// M303 E0 C8 S260
#define DEFAULT_Kp 15.37
#define DEFAULT_Ki 1.42
#define DEFAULT_Kd 41.54
```

Далее, проделываем аналогичную операцию для нагреваемого стола. Для этого заменяем «E0» на «E-1» и температуру меняем на 110°C.

Переносим «M303 E-1 C8 S110» в Pronterface, нажимаем «send» и ждём результатов. В редких случаях калибровка стола прерывается. Это происходит из-за того, что стол нагревается медленно и время ожидания выполнения команды истекает. В этом случае необходимо просто повторить отправку команды и процесс калибровки продолжится.

Полученные результаты переносим в код прошивки.

```
// M303 E-1 C8 S110
#define DEFAULT_bedKp 414.85
#define DEFAULT_bedKi 42.6
#define DEFAULT_bedKd 1008.67
```

3.1.2 Программно-инструментальная калибровка подачи филамента

После калибровки PID-контроллера температур необходимо приступить в калибровке подачи филамента. Необходимо обязательно убедиться, что Hot-End остыл, после заправляем пруток филамента в экструдер. Подшипник, в свою очередь, должен хорошо прижать его к зубчатому венцу. Вручную проверяем, чтобы пруток проталкивался хорошо.

Перед калибровкой осуществляются организационные меры, для последующей программной калибровки.

Кусачками отрезаем пруток ровно на входе в экструдер (Рисунок 23).

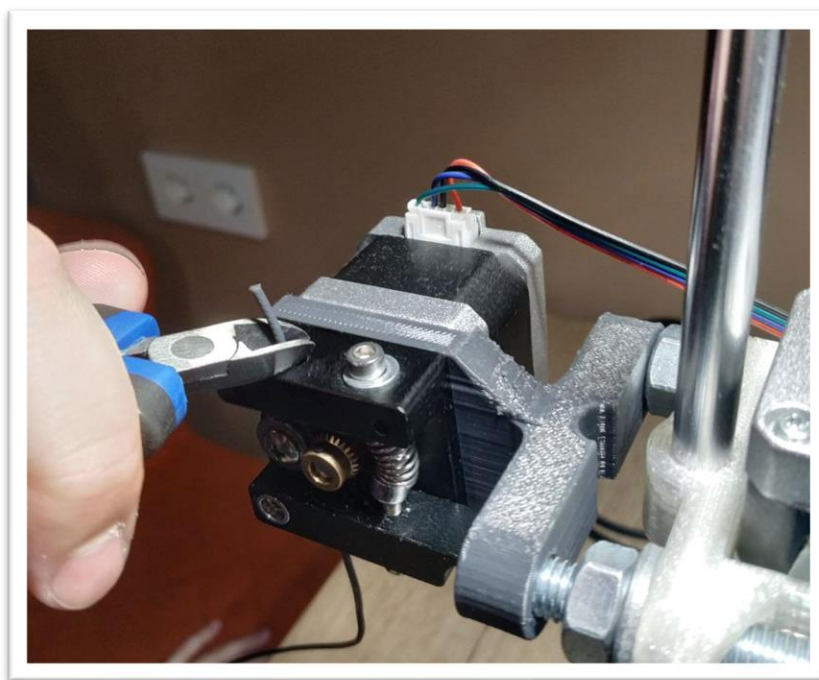


Рисунок 23 – Фиксирование прутка филамента

После переходим в программу «Pronterface», выставляем расстояние экструдирования 300 миллиметров (не рекомендуется ставить расстояние меньше 200 миллиметров, с целью повысить точность калибровки) (Рисунок 24).

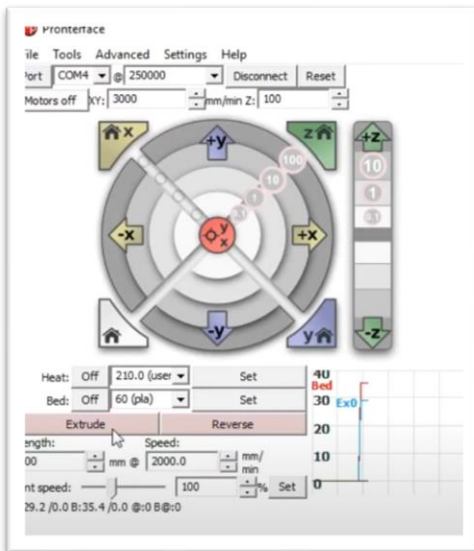


Рисунок 24 – Запуск экструдирования

Ждём, пока экструдер продавит пруток филамента. Помечаем маркером место входа прутка филамента в экструдер. Затем нажимаем кнопку «Reverse» в программе «Pronterface», не меняя значение расстояния (Рисунок 25).

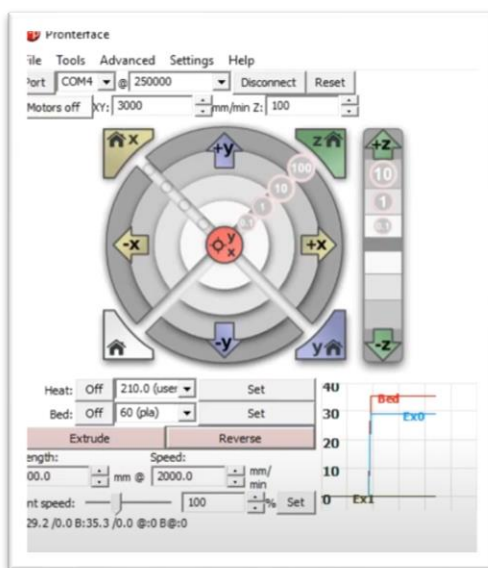


Рисунок 25 – Reverse экструдирования

Если метка вернулась в своё изначальное положение, значит экструдер держит пруток филамента достаточно крепко и нет проскальзывания.

Следует проверить еще и расстояние, на которое был сдвинут пруток. Обычной линейкой замеряем расстояние между 2-мя метками, которые оставили на прутке ранее.

Следуем алгоритму калибровки, указанному на рисунке 26:

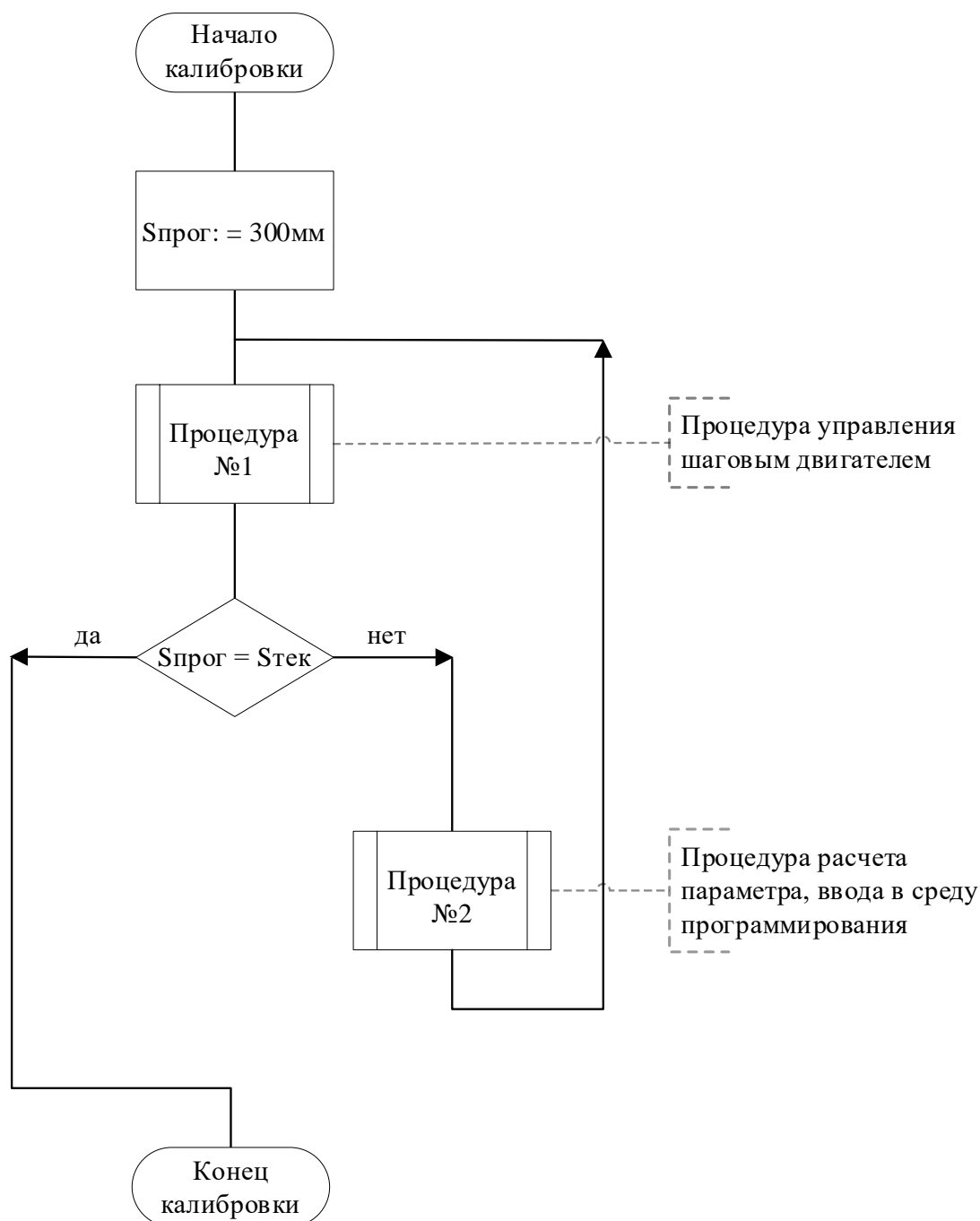


Рисунок 26 – Алгоритм программно-инструментальной калибровки

Если экструдер протолкнул пруток на то расстояние, которое было задано в «Pronterface», то количество шагов на проталкивание 1 миллиметра прутка посчитано и указано верно (Рисунок 27).

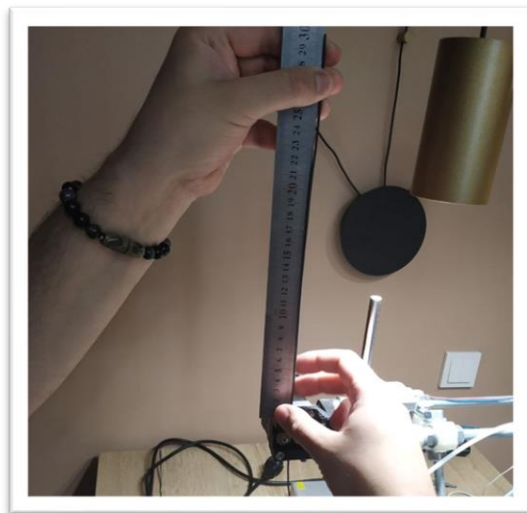


Рисунок 27 – Калибровка подачи филамента

В случае, если расстояние, заданное в программе, не сходится с полученным результатом, необходима коррекция шагового двигателя, отвечающего за подачу филамента.

Рассчитывается это следующим образом – необходимо вычислить новое значение по заданной формуле:

$$S_{\text{нов}} = S_{\text{тек}} * \delta, \quad (5)$$

где $S_{\text{нов}}$ – новое значение расстояния;

$S_{\text{тек}}$ – текущее значение;

δ – индекс погрешности.

В свою очередь, индекс погрешности считается по формуле:

$$\delta = \frac{S_{\text{жел}}}{S_{\text{факт}}}, \quad (6)$$

где $S_{\text{жел}}$ – желаемое расстояния;

$S_{\text{факт}}$ – фактическое расстояние.

К примеру, на данный момент у нас 96 шагов на подачу 1 миллиметра филамента экструдером, и мы хотим подать 300 миллиметров прутка, а по факту получилось 296

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 800, 96 }
```

Тогда получаем что индекс погрешности будет равен:

$$\delta = \frac{300}{296} = 1.01351351351,$$

Далее высчитываем новое значение:

$$S_{\text{нов}} = 96 * 1.01351351351 = 97,29729729696 .$$

Округляем до 97.3 и вносим в код прошивки, где выставляем количество шагов на 1 миллиметр вращения шагового двигателя экструдера. Далее повторяем калибровку с новыми значениями, чтобы убедиться в точности подачи филамента и повторяем эти действия, пока индекс погрешности не будет равен единице.

3.1.3 Подготовка тестовой модели

В качестве тестовой модели будет куб со сторонами 20 миллиметров [7].

Чтобы подготовить тестовый куб, воспользуемся программой «Fusion 360».

Fusion 360 – это программное обеспечение для 3D-моделирования, разработанное компанией Autodesk. Оно позволяет создавать, редактировать и

анализировать 3D-модели различных объектов, а также проводить расчеты и симуляции поведения этих объектов в различных условиях. Fusion 360 является интегрированным решением, которое поддерживает широкий диапазон операций 3D-моделирования, включая конструирование деталей, прототипирование и проектирование с оптимизированным расходом материала [10].

Далее обозначаем стороны X,Y,Z (Рисунок 28).

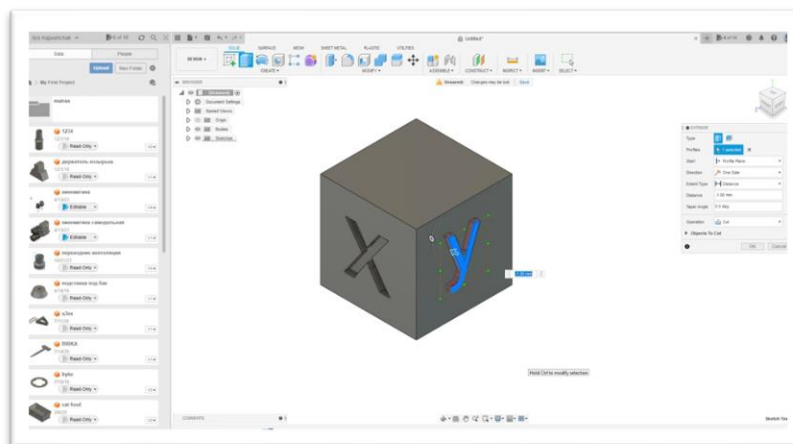


Рисунок 28 – Процесс подготовки тестового куба

После того, как все грани тестового куба будут готовы, проверяем, делаем небольшие углубления знаков X,Y,Z и сохраняем в формате STL (Рисунок 29).

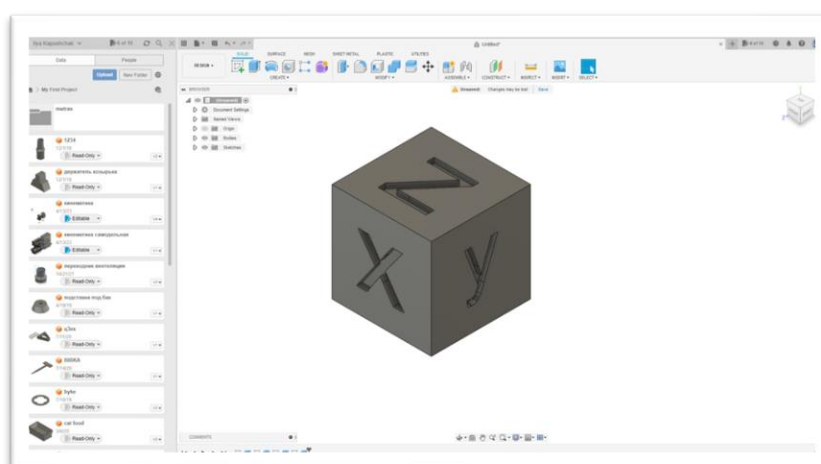


Рисунок 29 – Готовая модель тестового куба

После того, как модель тестового куба готова, необходимо загрузить модель в программу «UltiMaker Cura» [14].

UltiMaker Cura представляет из себя бесплатное программное обеспечение для 3D-печати, которое используется для настройки модели и создания файлов для печати на 3D-принтере. Оно разработано компанией «UltiMaker», предлагает широкий спектр функций, включая настройку слайсера, а также настройки для различных типов 3D-принтеров и филаментов.

UltiMaker Cura даст возможность нарезать тестовый куб на слои и увидеть продолжительность печати (Рисунок 30).

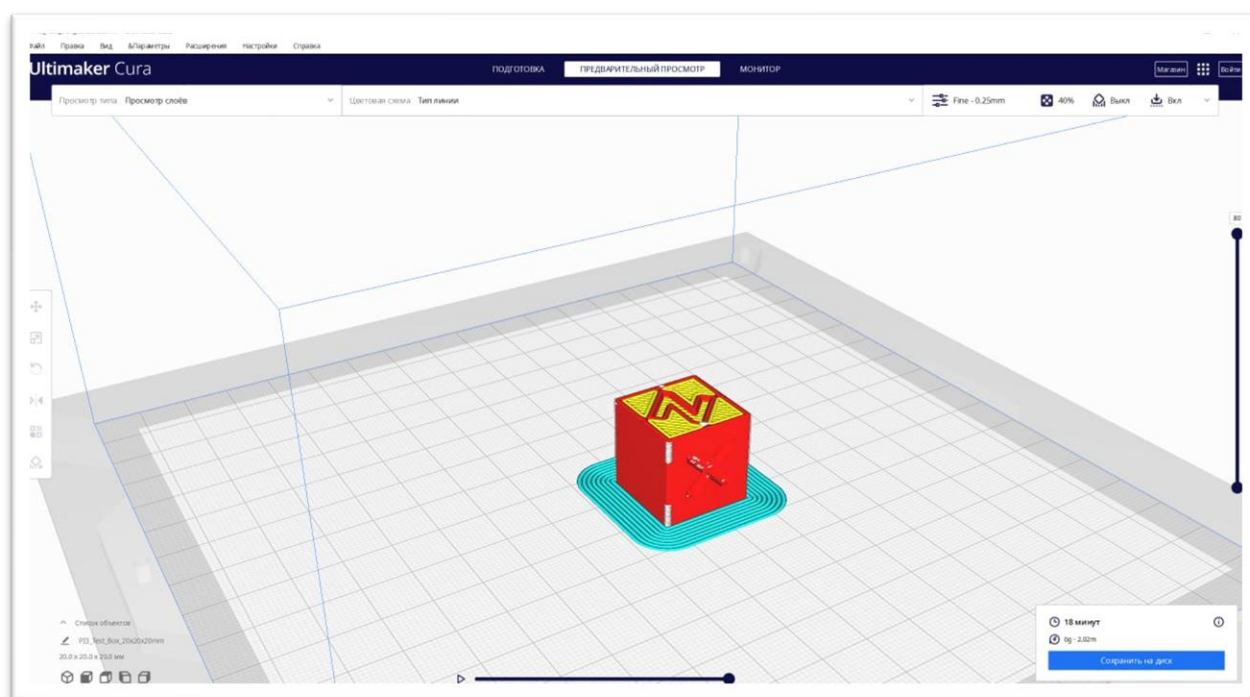


Рисунок 30 – Нарезка тестового куба на слои для печати

Также UltiMaker Cura дает возможность увидеть или поменять заполнение полости куба (Рисунок 31).

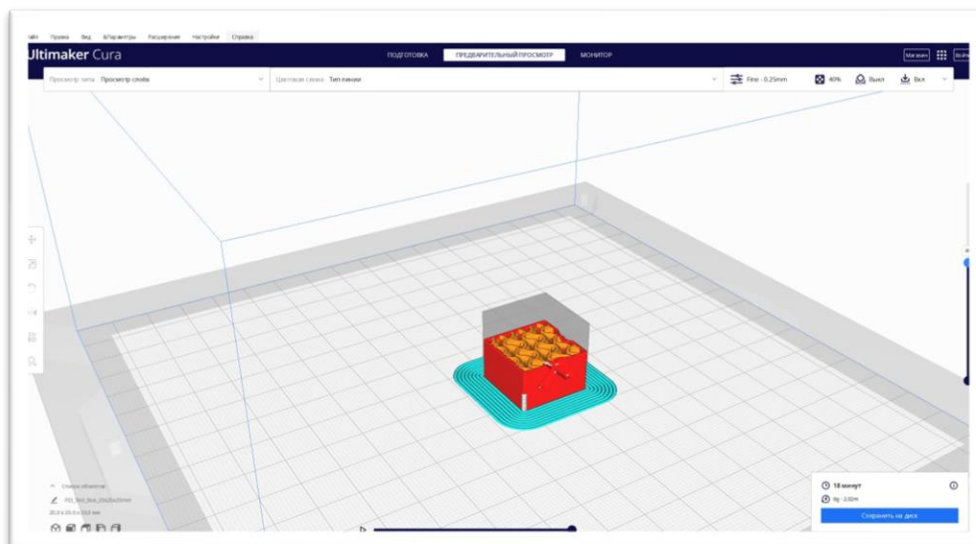


Рисунок 31 – Заполнение тестового куба

Обширные настройки дают возможность повлиять на прочность модели и на скорость печати (Рисунок 32).

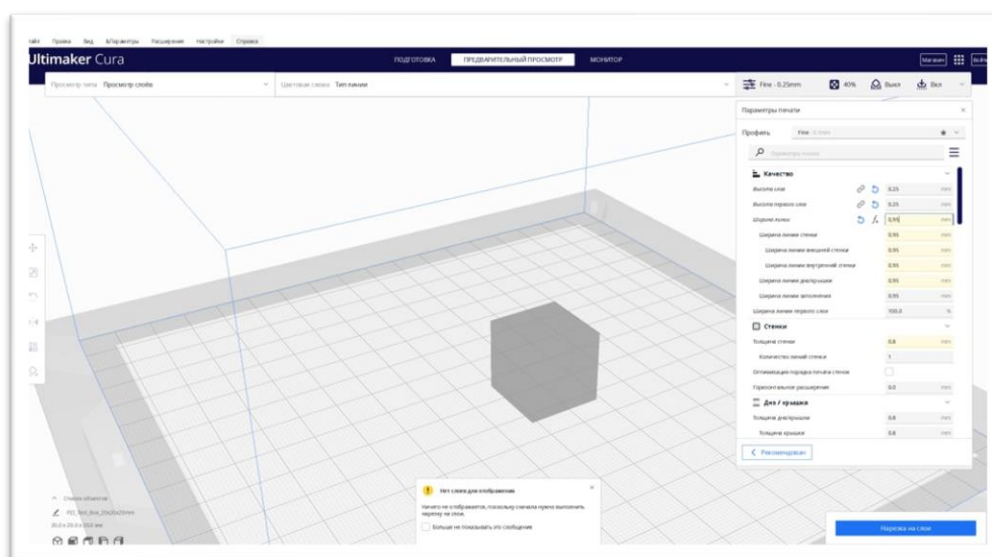


Рисунок 32 – Параметры заполнения тестового куба

После выбора настроек и нарезки на слои на выходе получается файл в формате «STL» – Test_Box_20x20x20mm.stl. В нем записан G-code, который необходимо загрузить в память дисплейного модуля с помощью SD-карты и запустить печать тестового куба.

3.1.4 Процесс реализации физической 3D-модели

После запуска печати идёт подготовка, которая включает в себя нагрев экструдера и стола. И как только нагревательный элемент обеспечит необходимую стартовую температуру, начинается процесс печати (Рисунок 33).

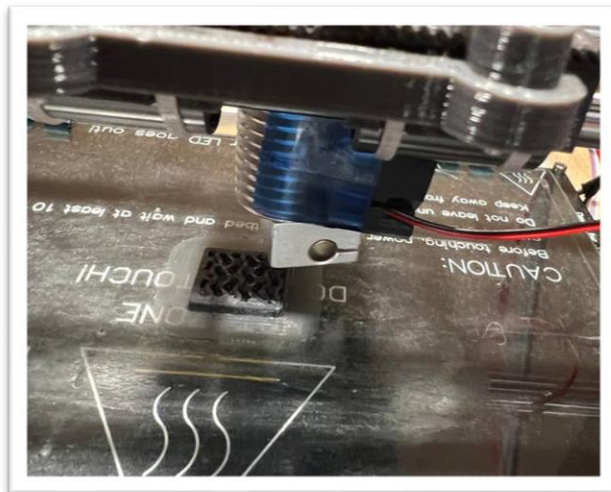


Рисунок 33 – Начало печати тестового куба

В зависимости от того, насколько высокое разрешение печати было задано в UltiMaker Cura, настолько быстро или долго будет процесс реализации 3D-модели. Но, чем выше разрешение, тем детальнее будет 3D-модель.

По завершению печати, экструдер выполняет команду «Home», это означает, что печать завершена, и модель готова к использованию (Рисунок 34).



Рисунок 34 – Готовая физическая 3D-модель тестового куба

И чтобы убедиться, что прошивка была произведена корректно, замеряем тестовый куб и смотрим, правильно ли каждый узел, прописанный в коде прошивки, выполнил все элементы в точности с заданными в коде параметрами.

По заданным параметрам 3D-модель тестового куба должна быть 20 на 20 миллиметров, что в итоге и получили.

Высокое разрешение можно достичь при высокой скорости печати и это напрямую зависит от заданных параметров 3D-модели и материала, используемого для печати.

Как правило, более высокое разрешение требует меньшей скорости печати, что дает более точные и детализированные результаты.

В нашем случае, был достигнут баланс между скоростью и точностью. Это дало возможность получить 3D-модель строго по заданным параметрам (Рисунок 35).

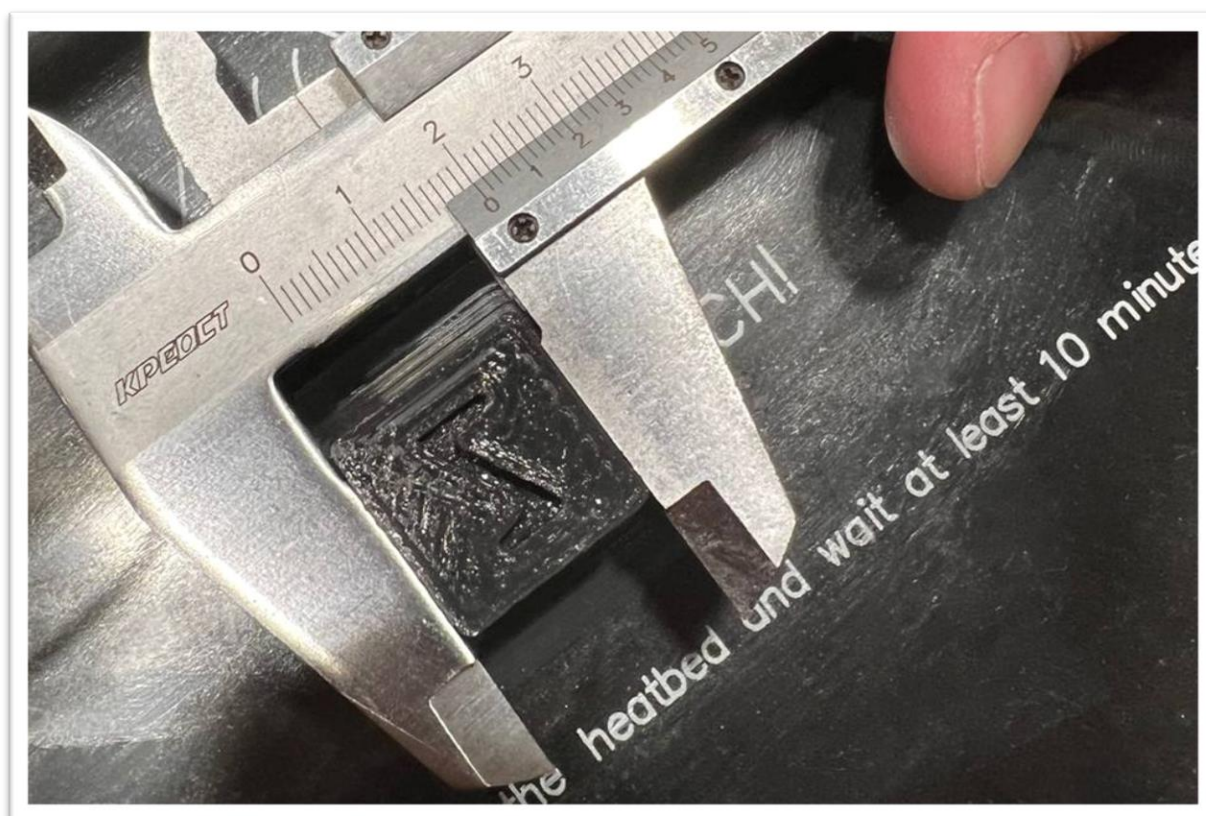


Рисунок 35 – Замер тестовой модели

ЗАКЛЮЧЕНИЕ

В результате ВКР была проделанная следующая работа:

1. Проведен анализ существующих аналогов аппаратно-программных комплексов для реализации физических 3D-моделей, обоснован выбор базы программного обеспечения для прототипа системы управления.

2. На основе анализа был осуществлен выбор среды разработки, также приведены структуры и описана последовательность действий программирования функциональных узлов прототипа системы управления на микроконтроллерной основе.

3. Проведено тестирование системы управления в виде эксперимента в выбранной среде программирования на действующем макете электромеханического агрегата при печати физической 3D-модели.

В результате всего вышесказанного можно утверждать, что поставленная цель была достигнута и задачи выполнены.

Полученный опыт и знания могут быть использованы для дальнейшего совершенствования 3D-комплексов и их интеграции в различные области промышленности, производства, дизайна, а также в учебных целях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аналоги Arduino IDE: сайт. – URL: <https://ruprogi.ru/software/arduino-ide> (дата обращения: 23.12.2022).
2. Обзор прошивки: сайт. – URL: <https://3deshnik.ru/blogs/akdzg/obzor-proshivki-teasur> (дата обращения: 12.02.2023).
3. Официальный магазин Microsoft: сайт. – URL: <https://apps.microsoft.com/store/detail/arduino-ide/9NBLGGH4RSD8?hl=ru-tm&gl=tm> (дата обращения: 10.01.2023).
4. Пайвин, А. С. Основы программирования станков с ЧПУ: учебное пособие / А. С. Пайвин, О. А. Чикова. – Екатеринбург: Уральский государственный педагогический университет: Институт физики, технологии и экономики Кафедра технологии и экономики, 2015. – URL: <http://elar.uspu.ru/bitstream/uspu/2887/1/uch00035.pdf> (дата обращения: 23.12.2022).
5. Программирование Ардуино: сайт. – URL: <https://arduino.ru/Reference> (дата обращения: 23.12.2022).
6. Прошивка для 3D-принтера 3D-ликбез: сайт. – URL: <https://3dtoday.ru/blogs/tiger/firmware-for-the-3d-printer-3d-educational-program> (дата обращения: 12.02.2023).
7. Тестовая модель: сайт. – URL: <https://www.thingiverse.com/thing:2119848> (дата обращения: 18.05.2023).
8. Что такое Arduino простыми словами // Амперка.ru : сайт-компания. – URL: <https://amperka.ru/page/what-is-arduino> (дата обращения: 23.12.2022).
9. 3D принтер RepRap Prusa Mendel // Arduino-diy.com : сайт сообщество – URL: <https://arduino-diy.com/arduino-3D-printer-RepRap-Prusa-Mendel> (дата обращения: 23.12.2022).
10. Fusion 360: сайт. – URL: <https://www.autodesk.com/products/fusion-360/overview?term=1-YEAR&tab=subscription&plc=F360> (дата обращения: 12.05.2023).

11. Github MarlinFirmware: сайт. – URL: <https://github.com/MarlinFirmware/Marlin> (дата обращения: 10.01.2023).

12. Github Pronterface: сайт. – URL: <https://github.com/kliment/Printrun> (дата обращения: 12.02.2023).

13. MarlinFirmware The code that makes the things: сайт. – URL: <https://marlinfw.org> (дата обращения: 10.01.2023).

14. UltiMaker Cura: сайт. – URL: <https://ultimaker.com/software/ultimaker-cura/> (дата обращения: 18.05.2023).

15. Visual Studio Code: сайт. – URL: <https://code.visualstudio.com/> (дата обращения: 23.12.2022).


Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ




Заведующий кафедрой

 О.В. Непомнящий
"21" 06 2023 г.

БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Аппаратно-программный комплекс для реализации физических
3D-моделей. Программная часть

Руководитель	 подпись	21.06.23 дата	доцент, канд. техн. наук должность, ученая степень	В.Г. Середкин
Выпускник	 подпись	21.06.23 дата		В.В. Юшков
Нормоконтролёр	 подпись	21.06.23 дата		В.Г. Середкин

Красноярск 2023