Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра вычислительных систем
Допустить к защите
зав. кафедрой д.т.н. доцент
_____ Курносов М.Г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Разработка пишущего принтера

Пояснительная записка

Студент: Антипова Е.П.

Факультет ИВТ Группа ИВ-621

Руководитель Гонцова А.В.

Новосибирск - 2020

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

КАФЕДРА
вычислительных систем

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА

СТУДЕНТУ Антиповой Е.П. ГРУППЫ ИВ-621

«УТВЕРЖДАЮ»
«__»_____
зав. кафедрой ВС
д.т.н. доцент
_____ Курносов М.Г.

Новосибирск, 2020 г.

1. Тема выпускной квалификационной работы бакалавра: «Разработка пишущего принтера» утверждена приказом СибГУТИ от «20» января 2020 г. № 4/54о-20

2. Срок сдачи студентом законченной работы: 12 июня 2020 г.

3. Исходные данные к работе

1 Документация по Arduino UNO

2 Плата Arduino UNO

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов)	**Сроки выполнения по разделам**
Постановка задачи, формирование плана работ	10.02.20-17.02.20
Изучение документации	17.02.20-24.02.20
Подбор средств разработки	24.02.20-10.03.20
Разработка плоттера	10.03.20-14.04.20
Разработка электрических схем плоттера	14.04.20-21.04.20
Разработка алгоритма для плоттера	21.04.20-15.05.20
Написание пояснительной записки	15.05.20-11.06.20

Дата выдачи задания: «__»_____

Руководитель _____ Гонцова А.В.

Задание принял к исполнению «__»_____

Студент _____ Антипова Е.П.

АННОТАЦИЯ

Выпускная квалификационная работа Антиповой Е.П.
по теме «Разработка пишущего принтера»

Объём работы 43 страниц, на которых размещены 29 рисунков и 10 таблиц. При написании работы использовалось 18 источника.

Ключевые слова: плоттер, arduino UNO, ЧПУ.

Работа выполнена на кафедре ВС СибГУТИ.
Руководитель – Гонцова А.В.,

Целью бакалаврской работы было разработать модель планшетного перьевого плоттера при помощи Arduino UNO, которая будет обеспечивать автоматическое рисование рисунков, чертежей и другой графической информации на бумаге формата А4.

В рамках бакалаврской работы была разработана и собрана модель планшетного перьевого плоттера при помощи Arduino UNO. Проверили на практике работоспособность графопостроителя, выполняющий автоматическое рисование чертежей на бумаге формата А4.

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

ОТЗЫВ

на выпускную квалификационную работу студента
группы ИВ-621 Антиповой Е.П.
по теме «Разработка пишущего принтера»

Графопостроитель, – это устройство, предназначенное для вывода различных чертежей, плакатов и других изображений на бумагу. Может использоваться в качестве альтернативы бытовому принтеру.

В рамках бакалаврской работы Антипова Е.П. разработала планшетный перьевой графопостроитель, позволяющий переносить изображения на бумагу формата А4.

В ходе выполнения работы студентка выбрала компоненты, разработала принципиальную схему и каркас, разработала программное обеспечение, собрала, протестировала и отладила устройство.

При выполнении работы Антипова Е.П. проявила самостоятельность, упорство, показала техническую грамотность, отличное владение практическими навыками.

Оригинальность текста составляет 95.75% Проверка осуществлялась в системе antiplagiat.ru.

Замечаний по работе нет.

Считаю, что бакалаврская работа заслуживает оценки «**отлично**», а Антипова Е.П. присвоения квалификации бакалавр по направлению 09.03.01 «Информатика и вычислительная техника», профиль «Электронно-вычислительные машины, комплексы, системы и сети».

Оценка уровней сформированности общекультурных,
общепрофессиональных и профессиональных компетенций обучающегося:

Компетенции		Уровень сформированности компетенций		
		Высокий	Средний	Низкий
Общекультурные	ОК-7 - способностью к самоорганизации и самообразованию	X		
Общепрофессиональные	ОПК-4 - способностью участвовать в настройке и наладке программно-аппаратных комплексов	X		
	ОПК-5 - способностью решать стандартные задачи профессиональной деятельности на основе информационной и	X		

	библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности			
Профессиональные	ПК-1 - способностью разрабатывать модели компонентов информационных систем, включая модели баз данных и модели и интерфейсов человек – электронно-вычислительная машина	X		
	ПК-2 - способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования	X		
	ПК-3 - способностью обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	X		

Работа имеет практическую ценность	X	Тема предложена предприятием	
Работа внедрена		Тема предложена студентом	X
Рекомендую работу к внедрению		Тема является фундаментальной	
Рекомендую работу к опубликованию		Рекомендую студента в магистратуру	X
Работа выполнена с применением ЭВМ	X		

Профессор кафедры вычислительных систем СибГУТИ

_____ Гонцова А.В.
(Гонцова Александра Владимировна)
«__»_____

Содержание

1 ВВЕДЕНИЕ

Графопостроитель(плоттер) – графическое устройство. Предусмотрен для рисования, вычерчивания любого графического материала (схема, чертеж) на бумаге, картоне, ткани и других подобных носителей. Является хорошей альтернативой бытового принтера.

Плоттеры применяются в полиграфической промышленности, в сферах, где необходимо получение конструкторских чертежей, архитектурных планов и других подобных схем, рисунков.

Виды плоттеров:

1. Режущие. Данный вид плоттера отличается тем, что может и печатать, и выполнять функцию резки. Может резать различные материалы.
2. Планшетные плоттеры. Носитель фиксируется на специальном столе и остается неподвижным во время печати. Движется печатающий элемент.
3. Рулонные плоттеры. Рулонная бумага движется на барабане, а изображение наносится на нее в процессе прохождения через головку.
4. Перьевые плоттеры. Электромеханическое оборудование векторного типа. Изображение пером, который может двигаться в двух направлениях.
5. Струйные плоттеры. Печать происходит путем нанесения большого количества точек с краской.
6. Электростатические плоттеры. Создается скрытое изображения на специальную бумагу, после к заряженным частицам носителя прилипает жидкая красная. Завершающим этапом просушка бумаги.
7. Лазерные и светодиодные плоттеры. При помощи лазерного луча или светодиодов наносится невидимое изображение, потом к заряженным частицам бумаги пристает тонер, который впоследствии запекается.
8. Плоттеры с прямым выводом изображения. Изображение наносится на термобумагу, пропитанную специальным раствором. При прохождении через «гребенку» в местах воздействия термических нагревателей бумага меняет свой цвет.
9. На основе термообработки. Принцип работы подобный с плоттером прямого вывода изображения. Главное отличие в том, что при прохождении термобумаги через «гребенку» между ними находится специальный цветной донор. Для печати цветного изображения процесс прохождения через гребенку повторяется необходимое количество раз. [1]

2 ПОСТАНОВКА ЗАДАЧИ

Цель выпускной квалификационной работы является разработка и сборка модели планшетного перьевого графопостроителя с использованием Arduino UNO, которая будет обеспечивать автоматическое рисование рисунков, чертежей и другой графической информации на бумагу формата А4.

Для достижения этой цели, необходимо изучить теоретический материал, подобрать необходимые инструменты и электронные компоненты, создать макет планшетного перьевого графопостроителя и проверить его работоспособность на практике.

3 ГРАФОПОСТРОИТЕЛЬ

3.1 Планшетный перьевой графопостроитель

В плоттере планшетного перьевого типа к рабочей поверхности фиксируется лист бумаги, по которому перемещается рисующее перо. Перо поднимает и опускает шаговый двигатель. В качестве пера - обычная шариковая ручка.

Рисунок формируется на листе бумаги в процессе перемещения опущенной шариковой ручки. Чтобы установить ручку в другом месте, она автоматически поднимается и перемещается в нужное место.

3.2 Сборка конструкции

В бакалаврской работе будет разработан планшетный перьевой плоттер размер чертежной поверхности будет предназначен для формата А4. На рисунке 3.1 представлена схема каркаса.

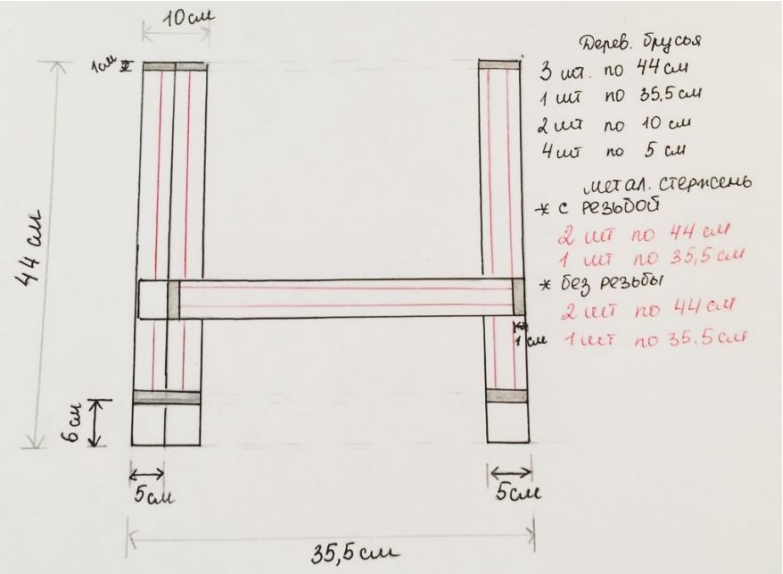


Рисунок 3.1. Конструктивная схема каркаса

4 ВЫБОР СРЕДСТВ РАЗРАБОТКИ

4.1 Arduino UNO

В качестве управляющего устройства используется Arduino UNO, представленная на рисунке 4.1.

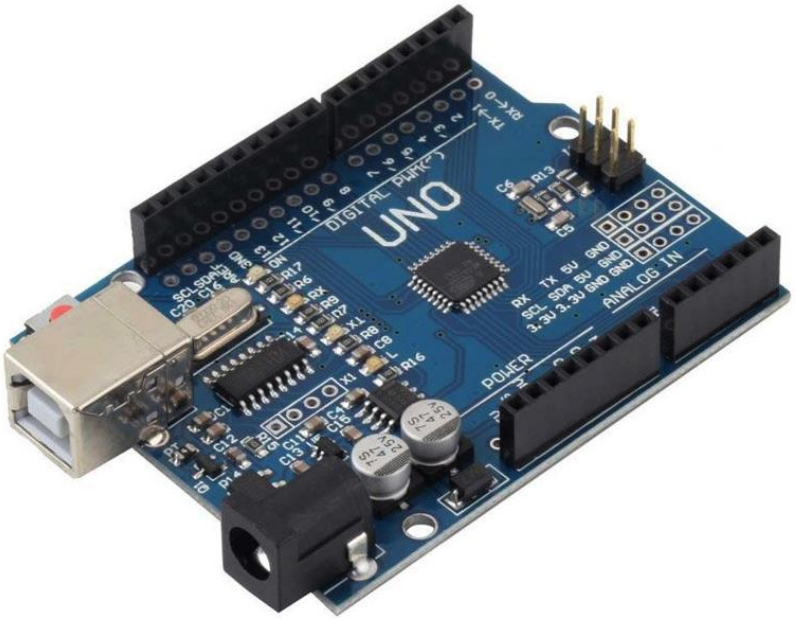


Рисунок 4.1 - Arduino UNO

Arduino UNO – контроллер, который построен на микроконтроллере ATmega328. В таблице 4.1 приведены технические характеристики Arduino UNO. Для работы необходимо подключить платформу к компьютеру с помощью USB-кабеля или подать питание при помощи батареи или адаптера AC/DC. Под топологию Arduino UNO имеется огромное количество плат дополнения (шилдов), которые обеспечивают дополнительную функциональность за счет их каскадного включения. Принципиальная схема приведена в приложении В.

Таблица 4.1 – Характеристики Arduino UNO

Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Напряжение питания	7-12 В
Напряжение питания	6-20 В
Цифровые входы/выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флэш-память	32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика
ОЗУ	2 Кб (ATmega328)
EEPROM	1 Кб (ATmega328)
Тактовая частота	16 МГц

4.1.1 Подключение к питанию

Рабочее напряжение платы Arduino UNO составляет 5 В. Так же на ней установлен стабилизатор напряжения, можно подавать на вход питание от разных источников. Источник питания выбирается автоматически.

- Через порт USB.
- Через контакт +5V. Он является и вводом, и выводом. Очень важно подавать ровно 5 В, иначе можно спалить сам микроконтроллер.
- Через штекер питания, расположенную на плате. Можно использовать батареи и блоки питания.
- Через контакт VIN. На плате установлен стабилизатор напряжения, через который проходит так от этого контакта. Рекомендуется использовать напряжение от 7 до 12 В.

4.1.2 Память

По умолчанию плата поддерживает три типа памяти:

1. Оперативная SRAM память (2 кБ), энергозависимая. Переменные и объекты хранятся по умолчанию.
2. Flash-память (32 кБ). Основное хранилище команд.
3. Энергонезависимая память (EEPROM) (1 кБ). Можно хранить данные, которые не исчезают после отключения питания.

4.1.3 Вводы и выводы

На рисунке 4.2 представлена плата с маркировкой выводов. [2]

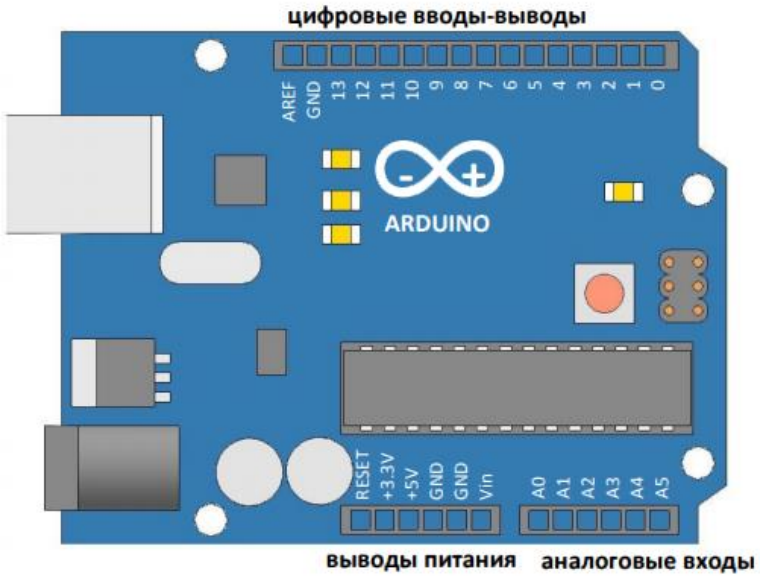


Рисунок 4.2 – Маркировка выводов платы Arduino UNO

4.1.3.1 Цифровые вводы-выводы

Каждый из 14 цифровых выводов может быть настроен как вход или выход. Каждый вывод имеет сопротивление нагрузки 20-50 кОм, а также может пропускать до 40 мА. Некоторые выводы имеют те же функции:

- Последовательная шина: 0 (RX) и 1 (TX). Выводы используется для приема (RX) и передачи (TX) данных TTL. Подключаются к выводам ATmega8U2 USB-to-TTL.
- ШИМ:3, 5, 6, 9, 10, 11. Обеспечивает ШИМ с 8-битным размещением.

- Внешнее прерывание: 2, 3. Могут быть сконфигурированы на вызов прерывания на переднем. заднем или младшем фронте, либо при изменении значения.
- LED: Встроенный светодиод, подключенный к цифровому выводу 13.
- SPI: 10(SS), 11(MOSI), 12(MISO), 13(SCK). Осуществляется связь SPI.
- AREF: опорное напряжение для аналоговых входов.

4.1.3.2 Аналоговые входы

Платформа имеет 6 аналоговых входов(A0-A5), каждый из которых имеет 10 бит и обычно имеют диапазон измерения до 5 В относительно земли.

- I2C: 4(SDA) и 5(SCL). Связь I2C(TWI) осуществляется с помощью выводов.

4.1.3.3 Выводы питания

Выводы питания, расположенные на плате:

- Vin: для внешнего источника питания.
- 5V: для питания микроконтроллера и компонентов на плате.
- 3.3V: выходное напряжение, генерируемое встроенным регулятором на плате.
- GND: вывод заземления.
- IREF: предоставляет платам расширения информацию о рабочем напряжении микроконтроллера Arduino UNO.
- Reset: низкий уровень сигнала перезагружает микроконтроллер.

4.2 Шаговые двигатели NEMA 17

С целью выполнения проекта было выбрано 3 шаговых двигателя формата NEMA 17. Текущий двигатель является гибридным, это означает, что он сочетает в себе характеристики шаговых двигателей и с переменным магнитным сопротивлением, и с постоянными магнитами.

Статор, на котором размещены обмотки возбуждения, гибридного двигателя имеет зубцы. Они обеспечивают, в отличие от основных полюсов, большое количество эквивалентных полюсов на которых размещены обмотки.

Полный оборот делится на 200 шагов по 1.8° и имеет рабочий ток до 1,7 А для каждой фазы. Двигатель представлен на рисунке 4.3. [3]

Шаговые двигатели позволяют достичь высокой точности позиционирования, по сравнению с другими типами двигателей. Вращение ротора двигателя обеспечивается переменным током на обмотке. Характеристики двигателя описаны в таблице 4.2.

ШД имеет две обмотки, по одной в каждой фазе, необходимая для изменения направления магнитного поля. На рисунке 4.4 представлены обмотки ШД Nema 17.

Рисунок 4.3 – Шаговый двигатель NEMA 17

Таблица 4.2 – Основные характеристики NEMA 17

Угол поворота за один шаг	1.8°
Диаметр вала	5 мм
Длина вала	24 мм
Длина мотора	40 мм
Ток на обмотку	1.7 А
Напряжение	5 – 24 В
Сопротивление обмотки	1.5 Ом
Крутящий момент удержания	4.2 кг/см
Количество контактов на разъеме	4
Масса	280 г

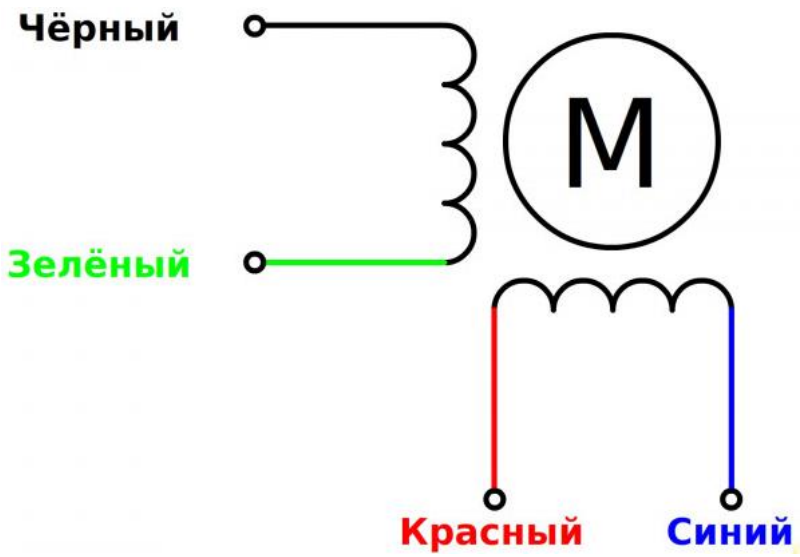


Рисунок 4.4 – Принципиальная схема шагового двигателя Nema 17

4.3 Шаговый двигатель 28BYJ-48

Шаговый двигатель 28BYJ-48 был выбран в данном проекте для того, чтобы поднимать и опускать шариковую ручку вместо сервопривода. На ось Z для удержания лучше подходят ШД, поскольку вал фиксируется в положении удержания совершенно неподвижно, а высокой скорости и ускорения по Z обычно не требуется. Сервопривод в таких условиях будет совершать микроколебания, что

нежелательно. Поэтому выбор пал на данный ШД, который представлен на рисунке 4.5. Принципиальная схема представлена на рисунке 4.6.

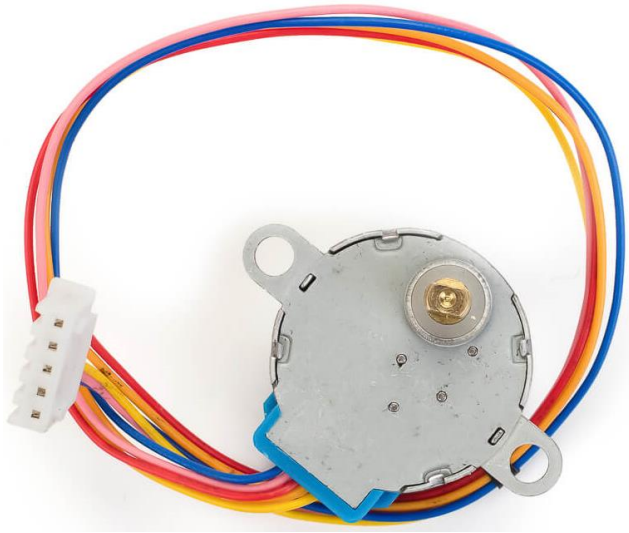


Рисунок 4.5 – Шаговый двигатель 28BYJ-48

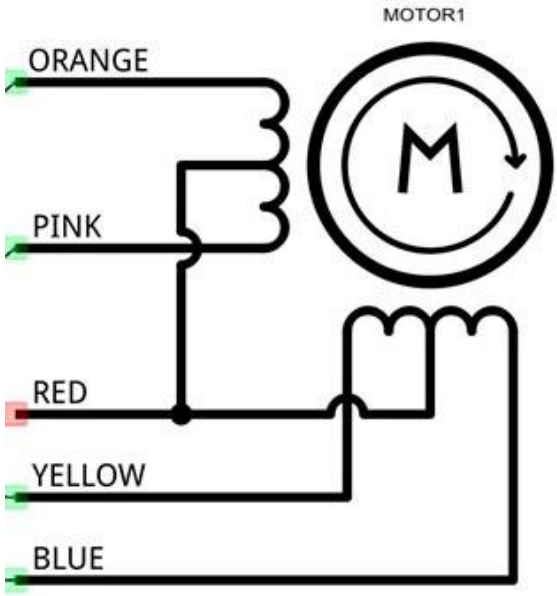


Рисунок 4.6 – Принципиальная схема шагового двигателя 28BYJ-48

Данный двигатель идеально подходит для управления шариковой ручки, так как он имеет очень маленький вес, в отличии от Nema 17 и не будет создавать лишнюю нагрузку. Характеристики шагового двигателя 28BYJ-48 представлены в таблице 4.3.

Таблица 4.3 – основные характеристики 28BYJ-48

	Полушаговый режим	Полношаговый режим
Напряжение питания	5 В	5 В
Число фаз	4	
Количество шагов за один оборот	64	32
Угол шага	5.625°	11.25°
Скорость вращения	15 оборотов в секунду	

Четырех фазный ШД – бесколлекторный двигатель. На роторе (валу) расположен магнит, а вокруг него – катушки. Если попеременно подавать ток на

эти катушки, то создается магнитное поле, которое отталкивает или притягивает магнитный вал, заставляя двигатель вращаться. Такая конструкция позволяет с большей точностью управлять валом, относительно катушек. [4]

Для того чтобы двигатель вращался по часовой стрелке, необходимо попеременно подавать напряжение на его выходы, в соответствии для полушагового и полношагового режимов, как это представлено на таблицах 4.4 и 4.5.

Таблица 4.4 – Фазы для полушагового режима

Контакт мотора	1	2	3	4	5	6	7	8
Оранжевый	+	+						+
Желтый		+	+	+				
Розовый				+	+	+		
Синий						+	+	+

Таблица 4. 5 – Фазы для полношагового режима

Контакт мотора	1	2	3	4
Оранжевый	+	+		
Желтый		+	+	
Розовый			+	+
Синий	+			+

4.4 Драйвер шагового двигателя A4988

В данной работе был выбран драйвер A4988, так как такой драйвер широко распространен и легко подключается. A4988 предназначен для управления биполярными шаговыми двигателями. Драйвер представлен на рисунке 4.7. Принципиальная схема представлена на рисунке 4.8.

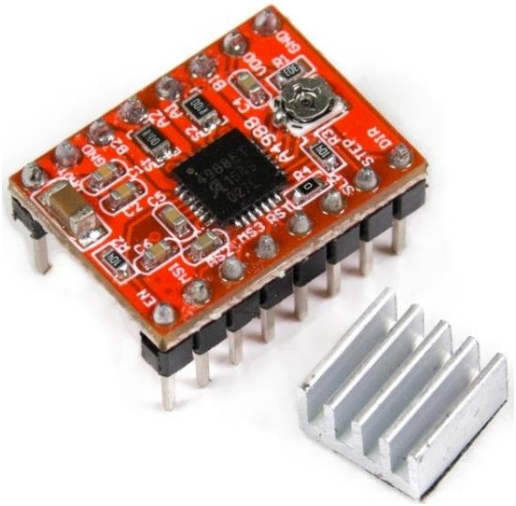


Рисунок 4.7 – Драйвер A4988 с радиатором охлаждения

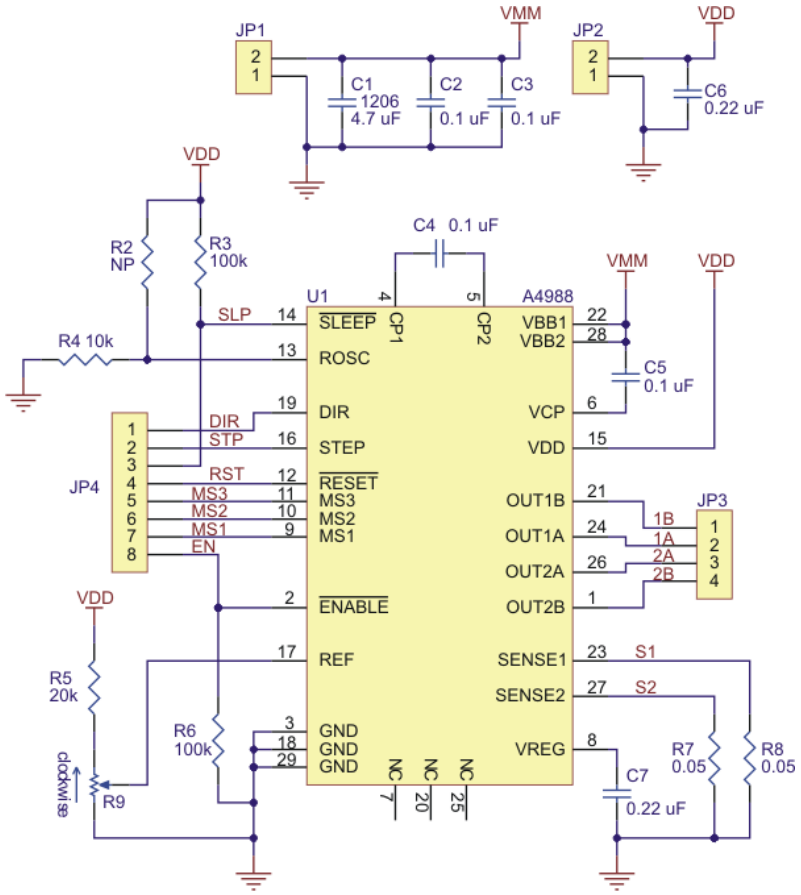


Рисунок 4.8 – Принципиальная схема драйвера A4988

Чип A4988 позволяет выбрать размер шага: 1, 1/2, 1/4, 1/8, 1/16. Он также оснащен большим количеством схем защиты, которые он отключает в случае: перегрева, перегрузки по току, короткого замыкания или малой мощности. Драйвер автоматически выбирает режим затухания тока (медленный/быстрый) и позволяет сохранить максимальный выходной ток двигателя. Основные характеристики A4988 описаны в таблице 4.6. [5]

Таблица 4.6 – Основные характеристики драйвера A4988

Напряжение питания	8 – 35 В
Режим микрошага	1, 1/2, 1/4, 1/8, 1/16
Напряжение логики	3 – 5.5 В
Защита от перегрева	
Максимальный ток на фазу (без радиатора)	1 А
Максимальный ток на фазу (с радиатором)	2 А
Размер	20 x 15 мм

Основной микросхемой модуля является драйвер биполярного шагового двигателя от Allegro – A4988, который имеет небольшие размеры, но позволяет работать с выходным напряжением до 35 В при токе до 1 А на катушку без радиатора и до 2 А с радиатором (дополнительное охлаждение). Для управления двигателями, нужны только два управляющих контакта: один для управления шагами, второй для управления вращения двигателя.

4.4.1 Назначение контактов драйвера A4988

На рисунке 4.9 представлены контакты драйвера A4988. [6]

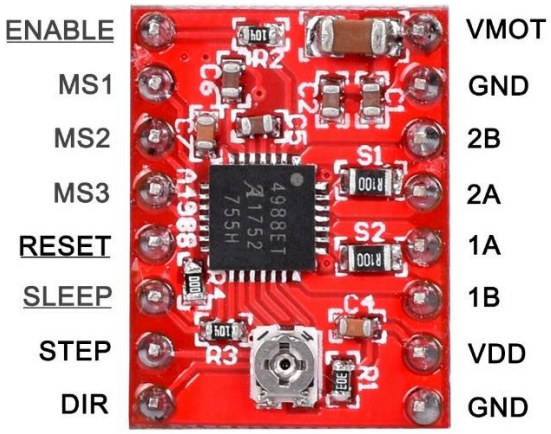


Рисунок 4.9 – Контакты драйвера A4988

На драйвере расположено 16 контактов:

1. ENABLE (EN) – позволяет работать чипу (0 – разрешить работу, 1 – отключить чип).
2. MS1, MS2, MS3 – выбор режима микрошаг.
3. RESET – перезагрузка логики чипа (0 – перезагрузить, 1 - обычный режим работы).
4. SLEEP – переключение чипа в спящий режим (0 – сон, 1 – обычный режим работы).
5. STEP – управляющий вывод, при каждом положительном импульсе, двигатель делает шаг (в зависимости от настройки микрошага), чем быстрее импульсы, тем быстрее вращается двигатель.
6. DIR – управляющий вывод, если подать +5 В двигатель будет вращаться по часовой стрелке, а при подаче 0 В – против часовой стрелки.
7. VMOT, GND – питание шагового двигателя от 8 до 35 В (обязательное наличие конденсатора на 100 мкФ).
8. 2B, 2A,1A,1B – подключение обмоток двигателя.
9. VDD, GND – питание внутренней логики от 3 В до 5.5 В.

4.4.2 Настройка микрошага

Драйвер может работать в микрошаговом режиме. Он может подавать питание на катушки со средними уровнями. Для двигателя Nema 17 с шагом 1.8 или 200 оборотов, в режиме 1/4, двигатель будет выдавать 800 шагов за оборот.

Размер шага задается комбинациями переключателей на входах (MS1, MS2, MS3). Благодаря им можно выбрать пять различных шагов, в соответствии с таблицей 4.7. На входы MS1 и MS3 переключатели установлены 100 кОм заземляющие резисторы, а на MS2 – 50 кОм. Если они остаются отключенными, двигатель запускается в полношаговом режиме. Для правильной работы в микрошаговом режиме необходим слабый ток, который обеспечивается ограничителями по току. В противном случае средние уровни будут неверно восприниматься, и двигатель будет пропускать микрошаги. [7]

Таблица 4.7 – Режимы микрошага

MS1	MS2	MS3	Микрошаг
Низкий	Низкий	Низкий	Полный шаг
Высокий	Низкий	Низкий	1/2 шага

Низкий	Высокий	Низкий	1/4 шага
Высокий	Высокий	Низкий	1/8 шага
Высокий	Высокий	Высокий	1/16 шага

4.4.3 Система охлаждения

Во время интенсивной работы микросхемы, A4988 начинает сильно нагреваться, и если температура превышает предельные значения, то может сгореть. Драйвер A4988 может работать с током до 2 А на катушку, но микросхема не греется если ток не превышает 1 А на катушку. Поэтому если ток выше 1 А необходимо установить радиатор охлаждения.

4.4.4 Настройка тока

Перед использованием двигателя необходимо сделать небольшую регулировку. Ограничить максимальный ток, протекающий через катушки ШД и ограничить его превышение номинального тока двигателя. Регулировка производится с помощью небольшого потенциометра. Регулировка производилась путем расчета значения напряжения V_{REF}, в соответствии с документацией на A4988 [8] вычисляется по формуле

$$I_{TripMAX} = \frac{V_{REF}}{8 * R_{SENSE}}, \quad (4.1)$$

где $I_{TripMAX}$– номинальный ток двигателя;

R_{SENSE}– сопротивление на резисторе.

Из формулы (4.1) можно получить необходимое значение напряжение:

$$V_{REF} = I_{TripMAX} * 8 * R_{SENSE}, \quad (4.2)$$

В нашем случае на драйвере A4988 установлены $R_{SENSE} = 0.100$ Ом (R100), на рисунке 4.10 обозначен синим цветом, а номинальный ток двигателя Nema 17 равен 1.7 А. Подставляем значения в формулу (4.2) и получаем значение V_{REF}.

$$V_{REF} = 1.7 * 8 * 0.1 = 1.36 \text{ В}$$

Это максимальное значение для двигателей. Но при таком напряжении двигатель будет нагреваться в режиме ожидания, нужно уменьшить это значение на 70%, то есть:

$$V_{REF} * 0.7 = 0.952 \text{ В}$$

С помощью отвертки и вольтметра устанавливаем необходимое значение. Установим плюсовой щуп вольтметра на потенциометр, а щуп заземления на вывод GND и выставим требуемое значение.

Для ШД 28DYJ-48 ток в 4-шаговом режиме равен 0.32 А. Вычислим по формуле 4.2 значение V_{REF} для этого двигателя, получим максимальное значение 0.256 В. Чтобы драйвер не перегревался умножим на 0.7 и получим необходимо значение для шагового двигателя 28BYJ-48, которое будет равно 0.18.

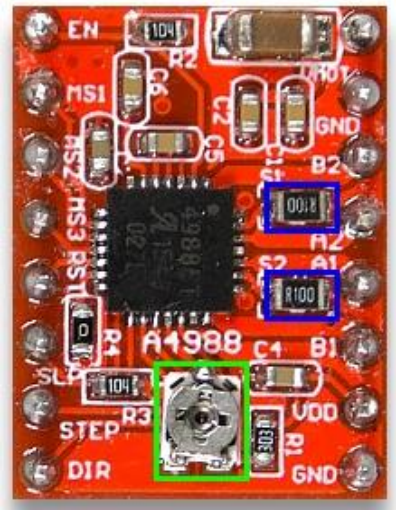


Рисунок 4.10 – Расположение деталей на плате драйвера A4988

Токочувствительные резисторы R_{SENSE} показаны синим цветом на рисунке 4.10, они равны 0.1 Ом. Поскольку драйвер ограничивает токи двух обмотках двигателя, то и резисторов R_{SENSE} на плате, тоже два. Зеленым цветом указан подстроечный резистор, для установки опорного напряжения V_{REF}.

4.4.5 Подключение драйвера A4988 к шаговому двигателю Nema 17

ШД Nema 17 имеет две обмотки. Определить принадлежность выходных концов обмоток можно с помощью их прозвонки мультиметром в режиме омметра. Один щуп оставить на произвольный вывод, а второй щуп – поочередно измерять активное сопротивление на всех остальных. Пара проводов, на которых будет обнаружено сопротивление в Омах, будет принадлежать одной обмотке.

После того, как узнали обмотки ШД. К двигателю необходимо подключить драйвер A4988: провода первой обмотки подсоединить к выводам 1A, 1B драйвера, провода второй обмотки к выводам 2A, 2B драйвера.

4.4.6 Подключение драйва A4988 к шаговому двигателю 28BYJ-48

ШД 28BYJ-48 является однополярным из-за соединения фаз, но в нашем случае драйвер A4988 предназначен для управления биполярными шаговыми двигателями. Поэтому необходимо модифицировать данный ШД.

Красный провод питания, не будет задействован и в схеме управления будет использовано 4 провода. Доработка заключается в разрыве дорожки, как это изображено на рисунке 4.11. Двигатель после доработки становится биполярным. [4]

Очень важно, при подключении к драйверу, поменять местами провода. То есть оранжевый и розовый подключить к выводам 1A и 1B, а синий и желтый к выводам 2A и 2B. В противном случае драйвер может сгореть.

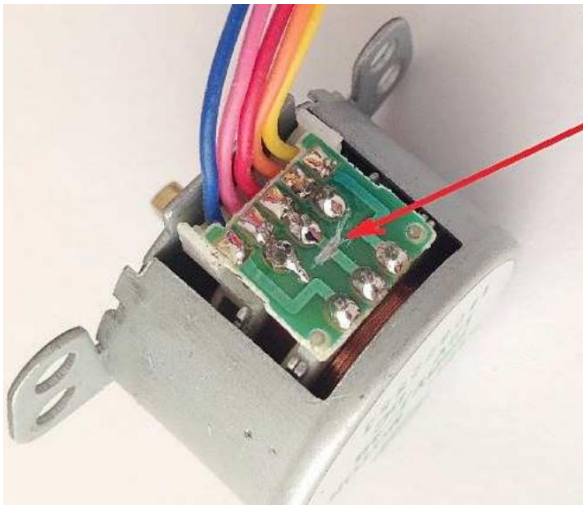


Рисунок 4.11 – Модернизация ШД 28BYJ-48

4.5 Плата расширения CNC Shield v3

Для создания модели мною была выбрана плата расширения CNC Shield v3, созданная на базе контроллера Arduino UNO. Набор электроники из Arduino UNO, CNC Shield v3 и драйверов шаговых двигателей позволяет управлять и реализовать параллельную работу шаговых двигателей. Так же плата дает возможность осуществлять проекты с двумя шаговыми двигателями на одну ось. [9]

На рисунке 4.12 представлена плата расширения.

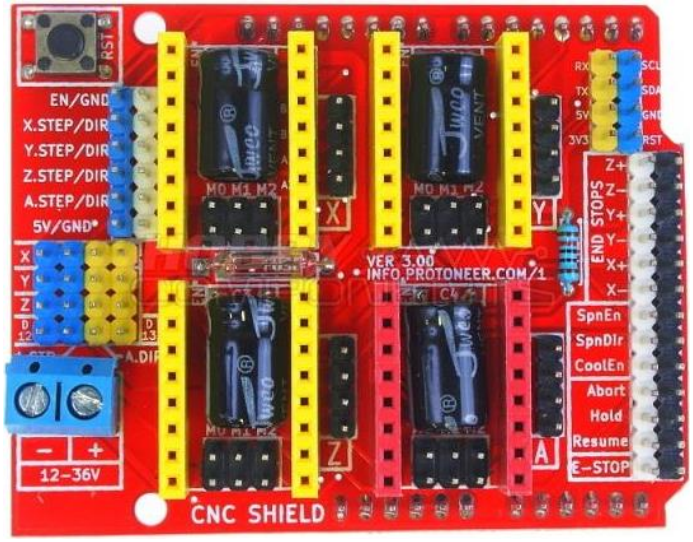


Рисунок 4.12 - Плата расширения CNC Shield v3

В таблице 4.8 описаны основные характеристики CNC Shield v3.

Таблица 4.8 – Характеристики CNC Shield v3

Количество осей	До 4 (X, Y, Z, A)
Концевые выключатели	До 6
Драйверы шаговых двигателей	A4988, DRV8825 или аналогичные
Интерфейсы	UART, I2C
Напряжение питания	12 – 36 В
Размеры	65x55x20 мм

Данная плата имеет 4 слота для подключения четырех драйверов двигателей. Слоты, показаны желтым цветом, отвечают за оси X, Y, Z, красный слот за ось A. Желтые и красный слоты представлены на рисунке 4.13.

4.5.1 Схема CNC Shield v3

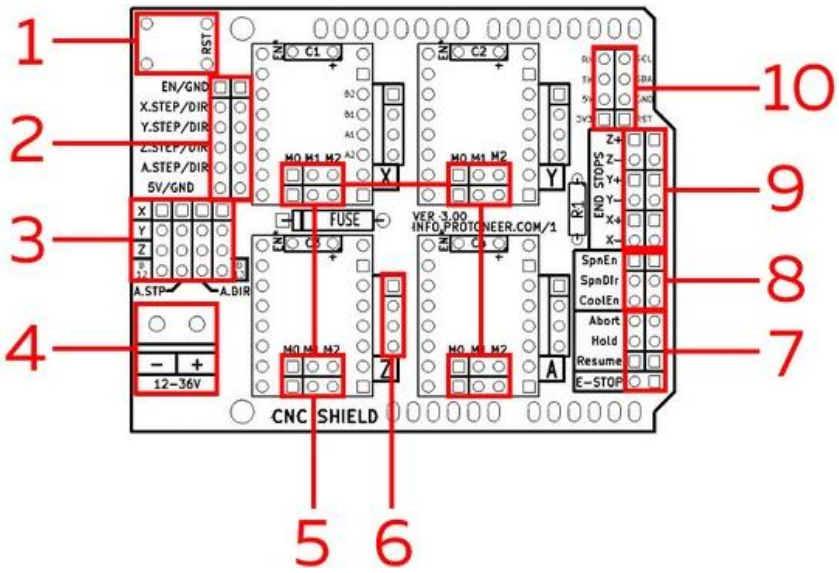


Рисунок 4.13 – Колодки платы CNC Shield v3

Числовые обозначения: [10]

1. Кнопка перезагрузки.
2. Колодка с контактами для подключения внешних драйверов ШД и контакты на 5 В и заземлению.
3. Колодка для дублирования осей. Ось A может дублировать одну из осей X, Y, Z, используя дополнительный драйвер и двигатель. То есть входной сигнал один и тот же, а драйверы и шаговые двигатели разные, но двигаются одинаково. Для реализации проекта с двумя ШД на одну ось необходимо установить 2 джемпера в соответствующие выводы. Колодку D12 необходимо замкнуть перемычками для управления шагом, а колодку D13 – для управления направлением вращения.
4. Разъем питания. На плату должны подавать питание 12 – 36 В.
5. Колодки управления микрошагом для драйверов ШД. Для работы необходимо вставить драйверы в слоты, настроить токи двигателей, согласно токам шаговых двигателей, установить перемычки в контакты M0, M1, M2, для определения режима работы драйвера и подключить к USB ПК.
6. Колодка для подключения шагового двигателя. Подключение шаговых двигателей происходит посредством разъемов на 4 контакта. Шаговый двигатель необходимо подключать в слот рядом с драйвером.
7. Кнопки и выключатели:
   - E-STOP: кнопка аварийной остановки.
   - Resume: кнопка продолжения.
   - Hold: кнопка паузы.
   - Abort: возврат на исходную позицию.
8. Разъемы для управления шпинделем и охлаждением.
   - CoolEn: включение подачи охлаждения.

- SpnEn: включение шпинделя.
- SpnDir: направление шпинделя.

9. Разъемы для подключения концевых выключателей. X+,X-,Y+,Y-,Z+,Z-. Необходимо соблюдать полярность при подключении концевиков.
10. Контакты интерфейсов UART и I2C.
    - Контакты для интерфейса UART: RT, TX, 5V, 3V3.
    - Контакты для интерфейса I2C: SCL, SDA, GND, RST.

4.5.2 Подключение питания

Помимо подключения к ПК кабелем USB необходимо подать силовое напряжение 12 В. На CNC Shield v3 это можно реализовать двумя путями:

1. Подключить блок питания с помощью разъема DC.
2. Подключить блок питания к клеммной колоде проводами.

4.6 Сервопривод Tower Pro MG90S

Сервопривод – вид привода, который может точно управлять параметрами движения. То есть, может повернуть свой вал на определенный угол или поддерживать непрерывное вращение с точным периодом.

Tower Pro MG90S миниатюрный сервопривод с металлическим редуктором. [11] Благодаря малым габаритным размеру и весы, сервопривод без особого труда поместиться на нашей модели для поднятия и опускания пера. На рисунке 4.14 представлен сервопривод.

Рисунок 4.14 - Сервопривод Tower Pro MG90S

При подключении к Arduino, лучше всего использовать внешние источники питания, так как у сервопривода Tower Pro MG90S повышенное потребление тока. В таблице 4.9 описаны основные характеристики Tower Pro MG90S.

Таблица 4.9 – Характеристики Tower Pro MG90S

Рабочее напряжение	4.8 – 6.08 В
Угол поворота	0 - 180°
Угловая скорость	0.1 сек/60°
Материал шестеренок	Метал
Управление	ШИМ
Рабочий ток	170 – 900 мА
Размеры устройства	22.5x12x35.5 мм

4.6.1 Управление сервоприводом

Главное в управление выполняет управляющий сигнал, который представляет собой импульсы постоянной частоты и переменной ширины. Поворот вала мотора на заданный угол достигается изменением длины импульса. Длина импульса – один из важных параметров, который определяет положение сервопривода. [12] Интервал между импульсами управления у данной модели составляет 20 миллисекунд.

Сервопривод, достигнув угла 180, вал окажет воздействие на ограничитель, а тот отдаст команду на выключение мотора. Следует обратить внимание на напряжение, которое используется при работе сервопривода. Если значение напряжения превышает допустимые значения из технических характеристик сервопривода, механические части могут выйти из строя или зубчатые колеса провернутся в положение превышающее максимальное значение и механизм не сможет продолжать движение в обратном направлении.

4.6.2 Подключение сервопривода к Arduino UNO

Сервопривод имеет три контакты, которые окрашены в разные цвета:

- Коричневый провод ведет к земле.
- Красный провод ведет к питанию +5 В.
- Желтый провод – сигнал управления, подключается к цифровому контакту.

4.7 Блок питания

Легкий и не подверженный коррозии алюминиевый корпус с защитой от короткого замыкания. На рисунке 4.15 представлен блок питания. В таблице 4.10 описаны основные характеристики блока питания.

Очень важно не подключать и не отключать двигатель, когда на контроллер подано питание.

Рисунок 4.15 – Блок питания 12 В, 3 А

Таблица 4.10 – Характеристики блок питания

Напряжение	12 В
Сила тока	3 А
Тип подключения	Клеммник
Светодиодная индикация	

5 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 Программное обеспечение Arduino IDE

В качестве программного обеспечения (ПО) была выбрана Arduino IDE. Это приложение, которое позволяет создавать программы в удобном текстовом редакторе, компилировать их в машинный код и загружать на все версии Arduino. [13] Код на языке Wiring, который частично совместим с C/C++. Arduino IDE представлена на рисунке 5.1.

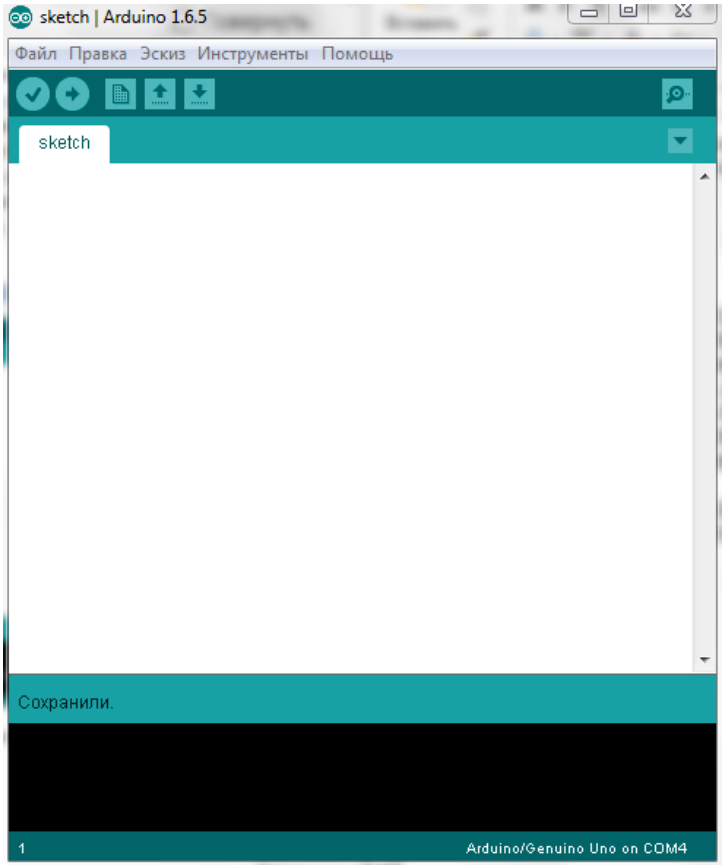


Рисунок 5.1 – Arduino IDE

Данная среда разработки включает в себя встроенный текстовый редактор, программный код, консоль, область сообщений и панель инструментов.

Скетч – программа, написанная в среде Arduino IDE, пишется в текстовом редакторе. Прежде чем начать работать необходимо соединить Arduino UNO с компьютером через USB-кабель, сообщить ПО номер драйвера виртуального COM-порта, который компьютер присвоил нашей Arduino UNO. Когда среда настроена и плата подключена можно спокойно загружать код на плату. [2]

Как только программа загружается на Arduino, она начинает свое независимое существование. Каждый раз, когда подаем питание на наш контроллер, эта программа будет автоматически запускаться с самого начала.

5.2 Программа для генерации G-code

G-code – общее название языка программирования устройств с ЧПУ. То есть это набор последовательных команд, который служит для сообщения станку куда ему нужно перемещаться и какие действия необходимо выполнить. [14]

Программы с G-кодом пишутся в виде текстового формата, каждую строчку называют кадром. Кадр состоит из буквенного символа – это адрес и цифра, в которой выражено числовое значение.

Программа генерации G-code позволяет генерировать G-код для фрезерных, лазерных и рисующих станков. На рисунке 5.2 представлена данная программа.

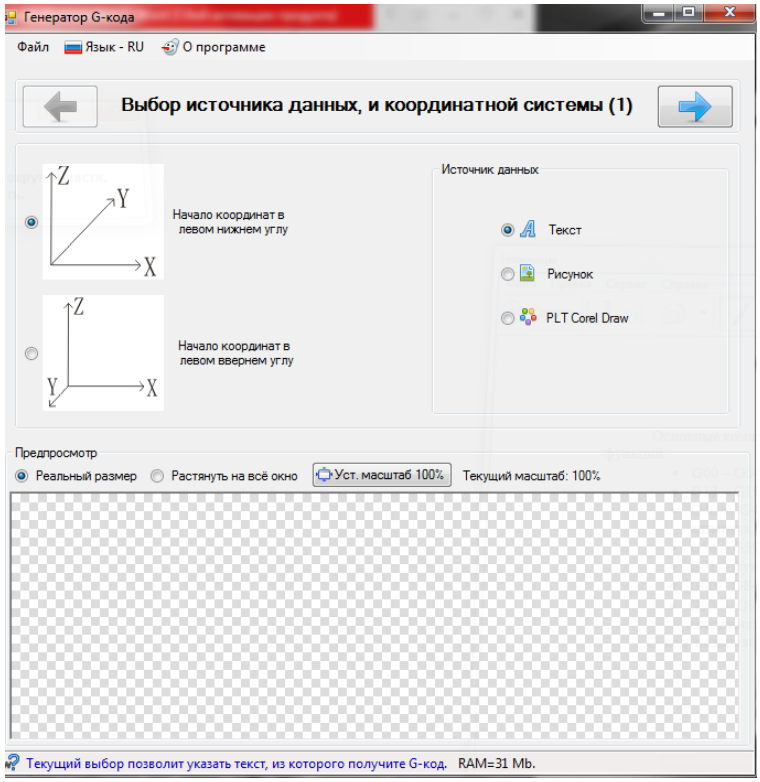


Рисунок 5.2 – Приложения для генерации G-кода

Источником данных могут быть:

- Введенный пользователем текст.
- Рисунок – анализ изображения и вычисляется контур изображения (jpg, jpeg, gif, bmp, png).
- Рисунок, выполняется генерация G-кода для выжигания лазером (jpg, jpeg, gif, bmp, png).
- PLT файл, получаемый из программы COREL DRAW.

Мы можем выбрать любой формат, для примера взяли текст, который вводится в самой программе. На последнем этапе, представленный на рисунке 5.3, мы устанавливаем высоту в миллиметрах, на которую необходимо поднять наше пишущее устройство, определяем скорости и нажимаем на «Генерация G-кода», правее появляется сгенерированный G-код по введенному тексту. Копируем и вставляем наш G-code в файл с расширением .gcode.

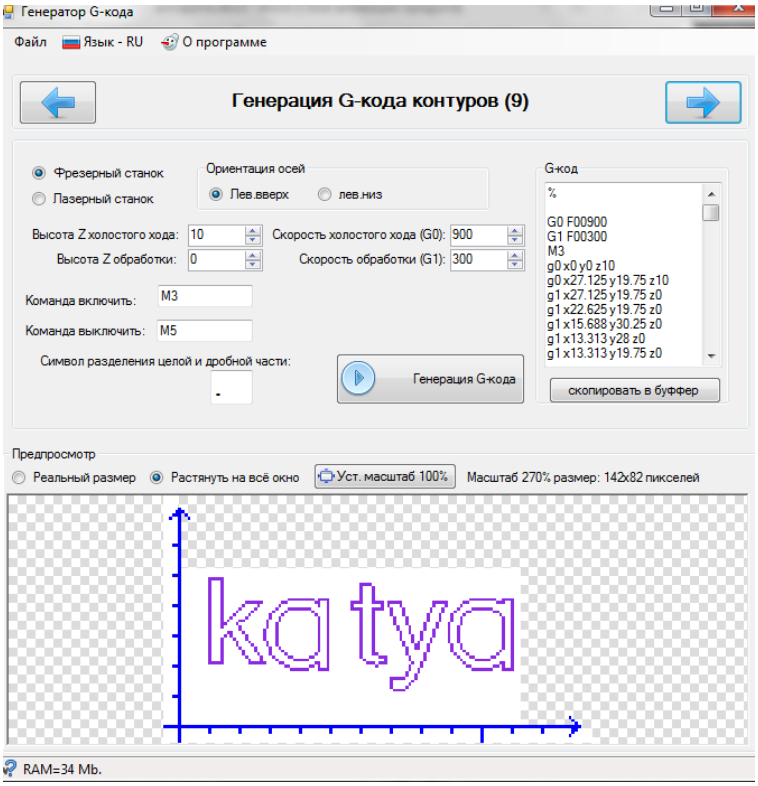


Рисунок 5.3 – Последний этап для генерации G-кода из введенного текста

5.3 Программное обеспечение Universal G-Code Sender

Universal G-Code Sender управляет ЧПУ станком, отправляя управляющий код (G-code команды) на контроллер.

Для работы с ней необходимо наличие Java 8. Подключает контроллер к USB-порту и указываем это в программе (COM4), устанавливаем скорость обмена – 115200 и подключаемся. [15] Данная программа представлена на рисунке 5.4.

Открывает файл с расширением .gcode, в котором находятся наборы команд и отправляем на чтение.

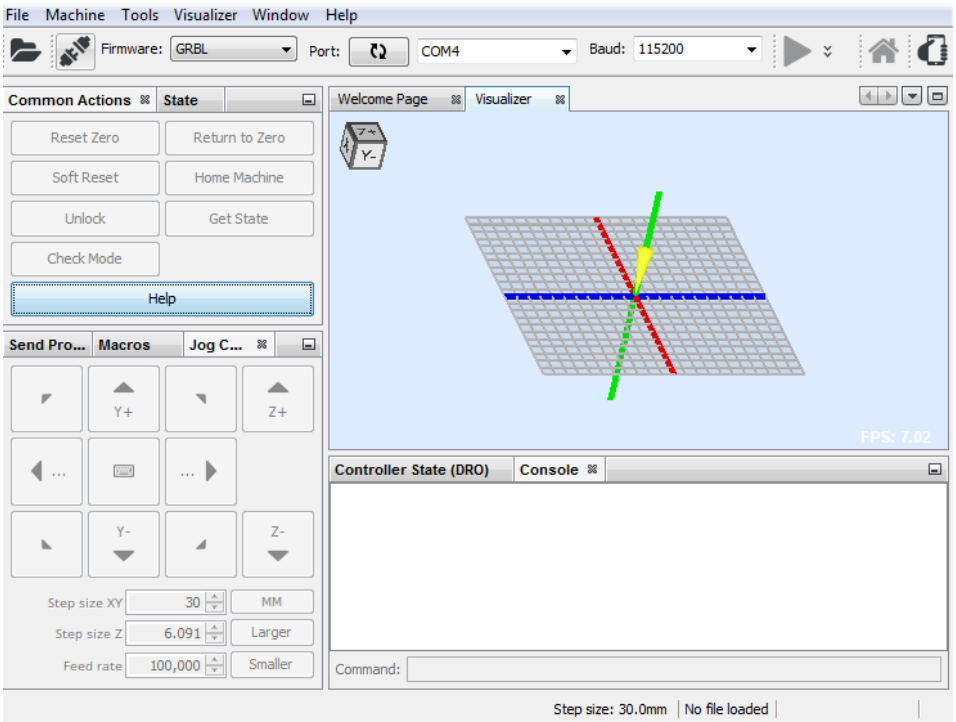


Рисунок 5.4 – Программное обеспечение Universal G-Code Sender

6 РАЗРАБОТКА ПРИНТЕРА

6.1 Сборка корпуса для графопостроителя

В качестве материала для каркаса были выбраны бруски дерева и металлические шпильки с резьбой и без резьбы. Сборка каркаса, представленного на рисунке 6.1, для плоттера принесла не мало проблем. Во-первых, было сложно соединить две оси, пришлось использовать двухкомпонентный эпоксидный клей, чтобы склеить деревянные брусья с гайками. Во-вторых, при склеивании, клей попадал на резьбу, тем самым затрудняя движения осей, проблему решили, используя смазку для шпилек с резьбой.

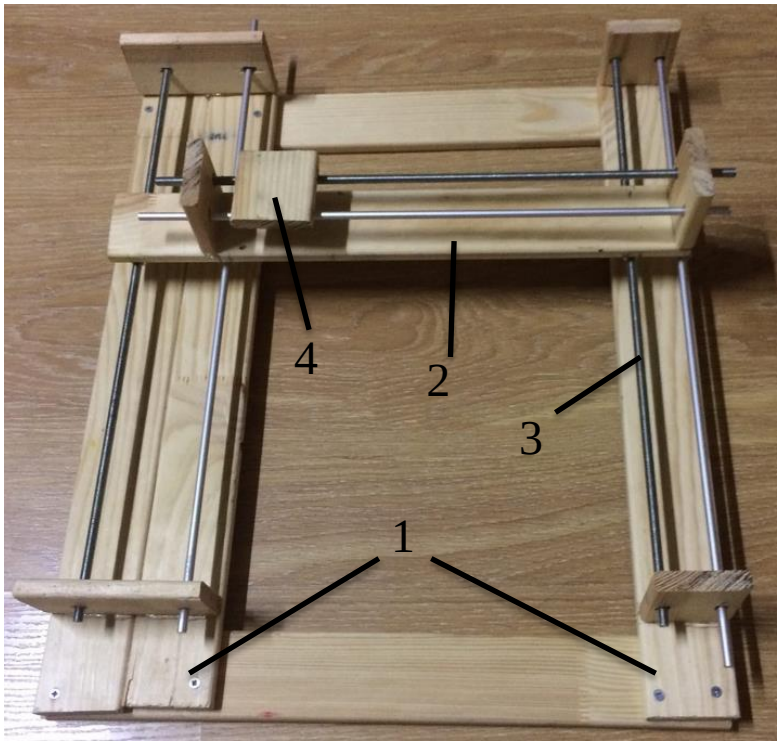


Рисунок 6.1 – Готовый каркас плоттера

Числовые обозначения:

1. Ось Y. К ним крепятся 2 шаговых двигателя, по одной на каждую, отвечающие за движение по оси Y.
2. Ось X. К ней крепится один шаговый двигатель, для перемещения по оси X.
3. Шпильки с резьбой. При вращении шаговых двигателей, вращаются шпильки. Гайки, закрученные на шпильки, двигают брусья осей X и Y.
4. Брусок, на который крепится 28BYJ-48 и пишущее устройство, в нашем случае это ручка.

6.2 Подключение электронных компонентов

После того, как каркас собран необходимо подключить все электронные компоненты друг с другом.

К плате Arduino UNO подключаем CNC Shield v3 вместе с драйверами, как это представлено на рисунке 6.2. в нашем проекте 4 драйвера, 3 из которых принадлежат осям X, Y, Z и один драйвер для дублирования оси Y.

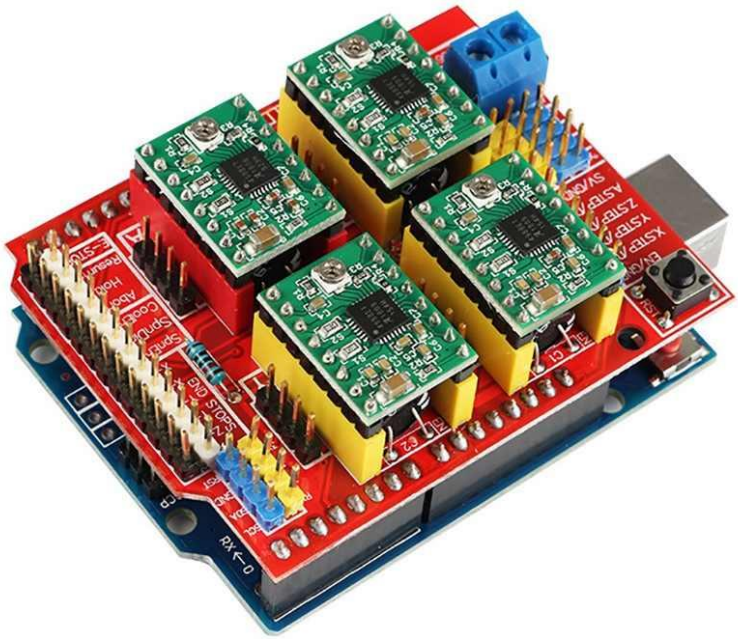


Рисунок 6.2 – Соединение платы Arduino UNO к плате расширения CNC Shield v3

Когда всю электронику соединили между собой, необходимо выставить опорное напряжение на драйверы. Опорное напряжение соответствует току подаваемую на двигатели. Чем выше напряжение, тем больше тока подается на двигатели и, тем больше мощность. Подключаем наш блок управления к ПК через USB-порт и с помощью вольтметра настраиваем необходимое напряжение для каждого драйвера. Для драйверов, к которым подключаем ШД Nema 17, необходимо напряжение составляет 0.95 В, а для ШД 28BYJ-48 напряжение равно 0.18.

После настройки опорного напряжения идет этап подключения ШД к плате управления, для этого используем провода, которые идут от ШД. На плате находим колодки и соединяем их с одноименными осями.

Далее используем блок питания на 12 В и 3 А и подключаем к плате, так же подключаю USB-кабель для связи с компьютером. Функциональная схема представлена на рисунке 6.3. Принципиальная схема подключения представлена в приложении Г.

Очень важную роль играет правильная последовательность подключения питания, так как при неправильном подключении она может сгореть. Поэтому сначала подаем питание 220 В от сети, а после подключает Arduino UNO.

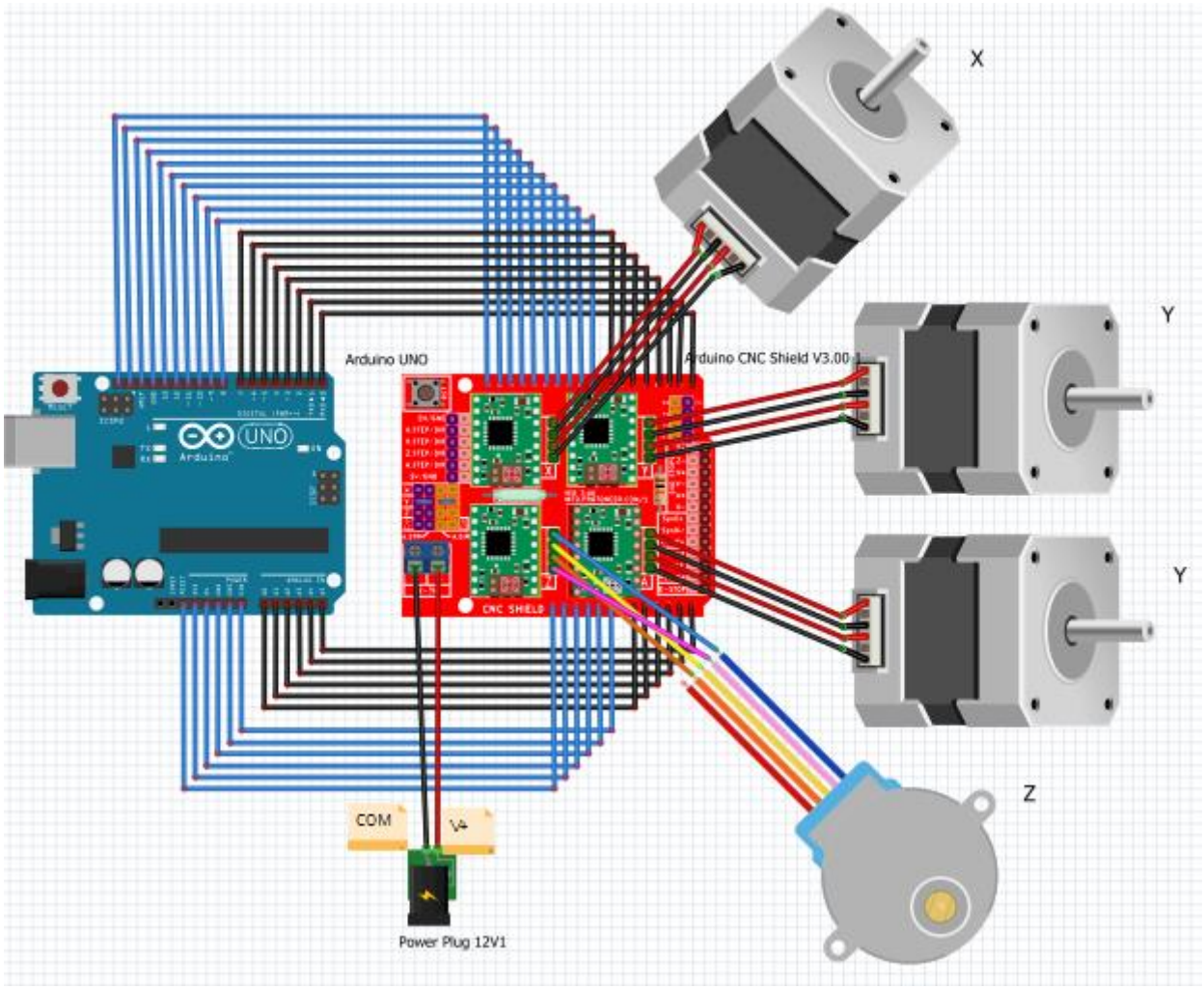


Рисунок 6.3 – Функциональная схема подключения плоттера

6.3 Работа с программным обеспечением

После сборки всей электроники, нам необходимо прошить плату Arduino UNO и проверить работу шаговых двигателей.

Открываем Arduino IDE и подключаем плату к компьютеру. Во вкладке инструменты выбираем плату «Arduino/Genuino UNO» и COM-порт к которому подключена плата.

В проекте используется библиотека GRBL. GRBL – это бесплатное ПО, написанное на языке С для управления ЧПУ станками, которое работает на прямую с Arduino UNO. Для работы с нашим плоттером пришлось ее модифицировать и добавить несколько функций.

Прежде чем откомпилировать и загрузить в Arduino UNO нашу модифицированную прошивку, нам необходимо ее добавить в библиотеку Arduino IDE, для этого надо общую папку с файлами, заархивировать и через вкладку скетч выбрать наш архив.

После заходим во вкладку файл, переходим в примеры выбираем GRBL и у нас открывается библиотека. Теперь необходимо загрузить ее в Arduino UNO. Нажимаем загрузить и ждем, когда скетч проверится и загрузится в плату. Видим, что загрузка завершена и плата у нас прошита и практически готова к работе с плоттером. На данном этапе можно проверить правильность загрузки прошивки в плату. Для этого заходим в программу Universal-G-Code-Sender и выбираем

скорость подключения к плоттеру, у нас она составляет 115200 бод (биты в секунду) и COM-порт, к которому подключена плата, как это представлено на рисунке 6.4. Нажимаем на кнопку Connect, если все правильно подключилось, то на консоли появится сообщение.

Рисунок 6.4 – Подключение платы управления в программе Universal-G-Code-Sender

Теперь можно проверить работу ШД в ручном режиме. Для этого используется панель управления, которая представлена на рисунке 6.5. Можно отрегулировать размер шага для осей X, Y и Z, а также установить скорость подачи. Если оси двигаются в противоположную сторону, то можно просто перевернуть разъем на плате.

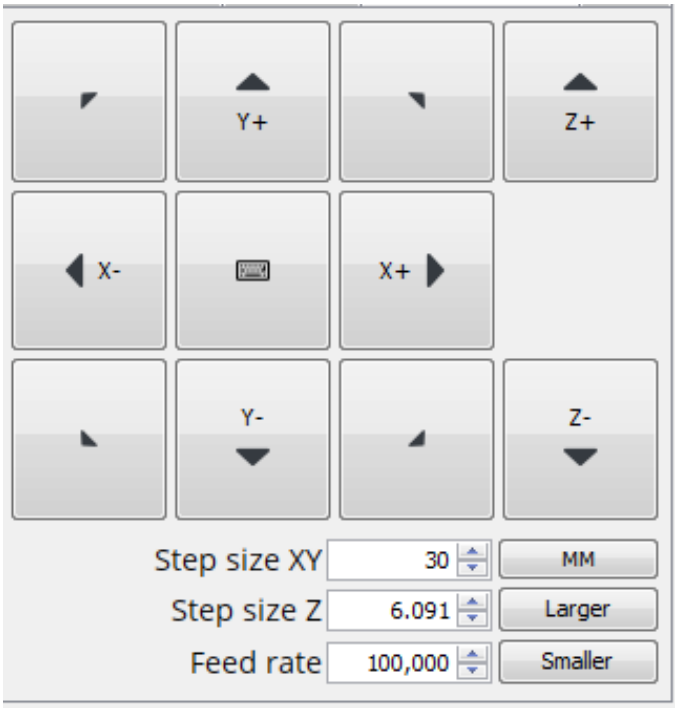


Рисунок 6.5 – Панель управления ШД в Universal-G-Code-Sender

В ручном режиме мы убедились в правильном подключении ШД к плате. Но нам надо программное управления, а для этого необходимо создать простой G-код для рисования. Для этого будем использовать программу генерации G-code. В программе выбираем что будем генерировать: текст или изображения. Для примера возьмем текст. Мы получаем из нашего текста сгенерированный G-code, представленный на рисунке 6.6.

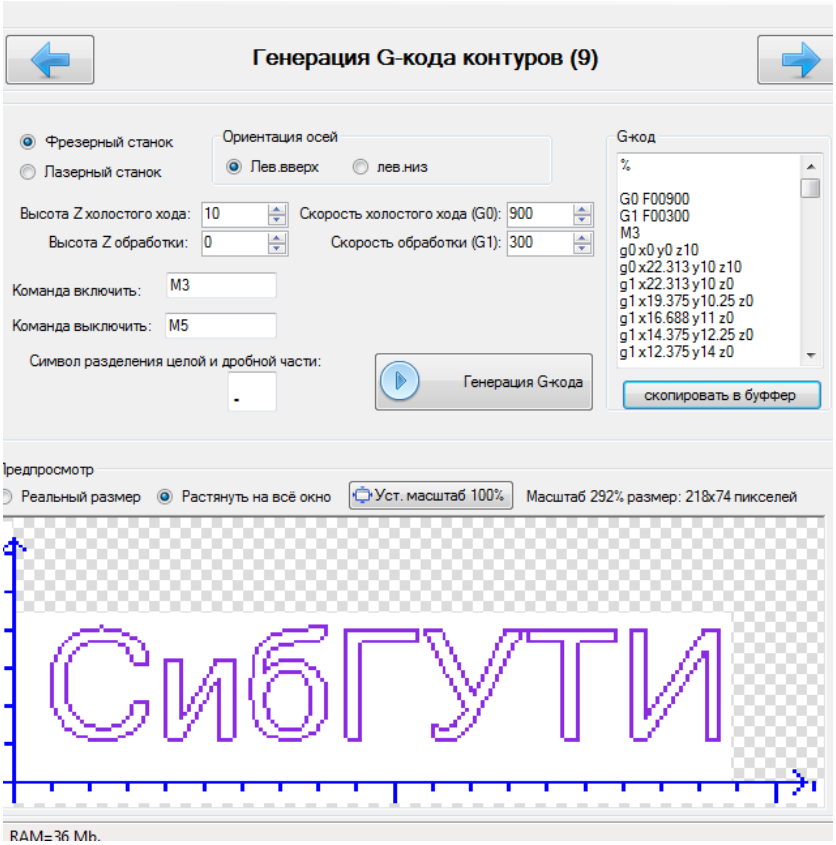


Рисунок 6.6 – Сгенерированный G-code из текста

Нажимаем на «скопировать в буфер», создаем текстовый файл с расширением .gcode и вставляем туда, полученный нами, G-код.

После этого, мы опять заходим в программу Universal-G-Code-Sender, нажимаем открыть файл, выбираем файл с расширением .gcode, который только что создали и открываем его в нашей программе. Наживаем на кнопку начать и видим, как наши двигатели начинают работать, а на экране мы можем наблюдать визуализацию работы станка. Визуализация представлена на рисунке 6.7.

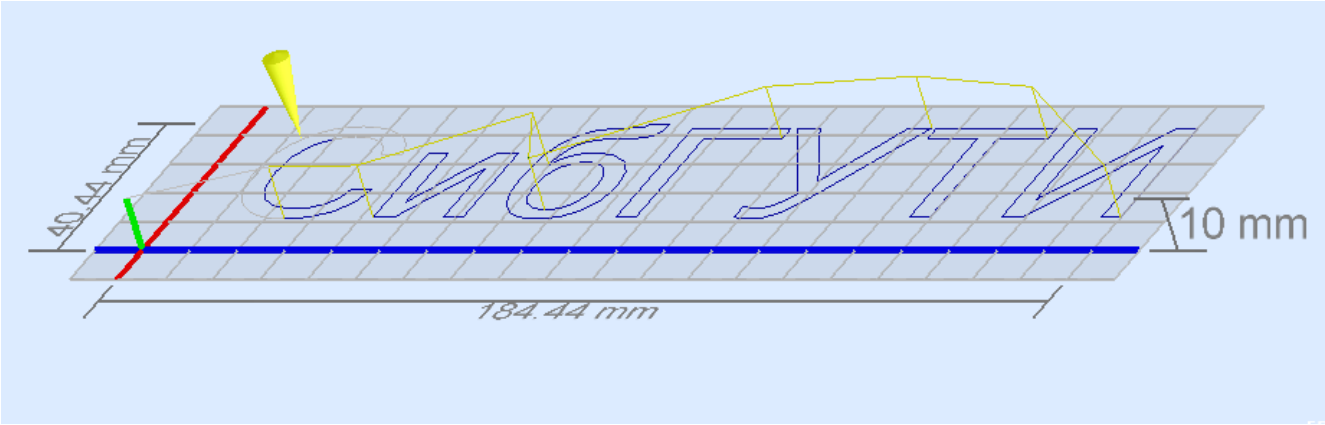


Рисунок 6.7 – Визуализация работы плоттера

7 АЛГОРИТМ РАБОТЫ

Основная работа с платформой Arduino UNO – это работа с форматом команд G-code. Для этого был реализован специальный парсер команд. Основные виды команд были вынесены в отдельный файл gcode.h в виде определенных директив препроцессора. Они представлены в листинге 7.1.

Листинг 7.1 – Основные виды команд

```
#define MODAL_GROUP_G0 0
#define MODAL_GROUP_G1 1
#define MODAL_GROUP_G2 2
#define MODAL_GROUP_G3 3
#define MODAL_GROUP_G4 4
#define MODAL_GROUP_G5 5
#define MODAL_GROUP_G6 6
#define MODAL_GROUP_G7 7
#define MODAL_GROUP_G8 8
#define MODAL_GROUP_G12 9
#define MODAL_GROUP_G13 10

#define MODAL_GROUP_M4 11
#define MODAL_GROUP_M7 12
#define MODAL_GROUP_M8 13
```

Режимы устанавливает анализатор в определённое состояние. Эти режимы объединены в группа, называемые «модальные группы», которые не могут быть логически активными одновременно, например, группа модальных единиц устанавливает интерпретируется ли G-код программы в дюймах или в миллиметрах. [16]

- G1 – [G0,G1,G2,G3,G38.2,G38.3,G38.4,G38.5,G80] Режим движения.
- G2 – [G17,G18,G19] Выбор плоскости.
- G3 – [G90,G91] Режим расстояния.
- G4 – [G91.1] Режим расстояния дуга IJK.
- G5 – [G93,G94] Режим скорости подачи.
- G6 – [G20,G21] Режим единицы.
- G7 – [G40] Режим компенсации радиуса резца. G41 / 42 не поддерживается.
- G8 – [G43.1,G49] Смещение длины инструмента.
- G12 – [G54,G55,G56,G57,G58,G59] Выбор системы координат
- G13 – [G61] Режим точного останова
- M4 – [M0,M1,M2,M30] Остановка, программный режим.
- M7 – [M3,M4,M5] Состояние шпинделя.
- M8 – [M7,M8,M9] Статус системы охлаждающей жидкостью.

Система Arduino UNO понимает команды, написанные в формате G-code. И для работы с ним необходим алгоритм разбора команд.

Строка команды, которая будет выполнена, должна состоять из букв в верхнем регистре и дробных положительных чисел. В листинге 7.2 приведен пример G-code.

Листинг 7.2 – Пример G-code

```
%
G0 F00900
G1 F00300
M3
G0 X34.063 Y19.75 Z10
G1 X10 Y19.75 Z0
M5
%
```

Разбор строки происходит по принципу поиска символа команды и следующим или следующими за ним цифровыми значениями, корректно записанными в формате языка G-код. Определение каждой группы команд выделено в отдельный блок обработки, например, раздельные блоки для команд вида GXXX и MXXX. Строка G-кода, которые пришли с внешнего устройства, преобразуясь, хранятся в структуре.

После парсинга текстовой команды с внешнего приложения, команда перед отправлением на выполнение производит дополнительную проверку корректности, устанавливая в случае необходимости пропущенные дополнительные параметры. Большинство параметров при формировании блока парсинга берут свои числовые значения из заголовочного файла gcode.h. Время выполнения разбора линейное O(n), так как проход в общем случае занимает всю строку от первого символа до последнего.

Работа с парсингом G-code проходит в 3 этапа.

На первом этапе происходит инициализация структур в файле gcode.h для хранения и обработки команд G-code, которые проходят через внешнее управление. В листинге 7.3 представлен пример кода.

Листинг 7.3 – Пример инициализации структур

```
memset(&gc_block,0, sizeof(parser_block_t));
memcpy(&gc_block.modal,&gc_state.modal,sizeof(gc_modal_t))
;
  uint8_t axis_command = AXIS_COMMAND_NONE;
  uint8_t axis_0, axis_1, axis_linear;
  uint8_t coord_select = 0;
  float coordinate_data[N_AXIS];
  float parameter_data[N_AXIS];

  uint8_t axis_words = 0;
  uint8_t ijk_words = 0;

  uint16_t command_words = 0;
  uint16_t value_words = 0;
```

На втором этапе происходит обработка полученной строки. Обработка происходит до тех пор, пока не получат терминальный символ строки \0. В листинге 7.4 представлен пример кода.

Листинг 7.4 – Пример обработки строки

```
switch(letter) {
```

```
        case 'G':
          switch(int_value) {
            case 10: case 28: case 30: case 92:
              if (mantissa == 0) {
                if           (axis_command)           {
FAIL(STATUS_GCODE_AXIS_COMMAND_CONFLICT);           }
axis_command = AXIS_COMMAND_NON_MODAL;
              }
```

Происходят разбор команд и проверка на наличие нарушений модальных групп. Идет анализ строки, символ за символом и привязывается к внутренней структуре parser_block. Где, в случае успеха, будет содержаться правильная корректная команда G-code для работы Arduino UNO.

На третьем этапе происходит повторная обработка ошибок из первого шага. Ошибки вида «подана команда для работы с осью, но не указана сама ось». В листинге 7.5 представлен пример кода. Исправляется путем подстановки стандартных значений.

Листинг 7.5 – Пример повторной обработки ошибок

```
if (axis_words) {
    if(!axis_command)       {       axis_command       =
AXIS_COMMAND_MOTION_MODE; }
  }

  if (bit_istrue(value_words,bit(WORD_N))) {
    if    (gc_block.values.n    >    MAX_LINE_NUMBER)    {
FAIL(STATUS_GCODE_INVALID_LINE_NUMBER); }
  }
```

Основной цикл работы, заключается в функции protocol_main_loop (), который представленный в листинге 7.6.

Листинг 7.6 – функция protocol_main_loop()

```
void protocol_main_loop()
{
  report_init_message();
  if (sys.state == STATE_ALARM) {
    report_feedback_message(MESSAGE_ALARM_LOCK);
  } else {
    if (system_check_safety_door_ajar()) {
      bit_true(sys_rt_exec_state, EXEC_SAFETY_DOOR);
      protocol_execute_realtime();
    } else {
      sys.state = STATE_IDLE;
    }
    system_execute_startup(line);
  }

  uint8_t comment = COMMENT_NONE;
  uint8_t char_counter = 0;
  uint8_t c;
  for (;;) {
```

```
    while((c = serial_read()) != SERIAL_NO_DATA) {
      if ((c == '\n') || (c == '\r')) {
        line[char_counter] = 0;
        protocol_execute_line(line);
        comment = COMMENT_NONE;
        char_counter = 0;
      } else {
        if (comment != COMMENT_NONE) {
           if (c == ')') {
            if (comment == COMMENT_TYPE_PARENTHESES) {
comment = COMMENT_NONE; }
          }
        } else {
          if (c <= ' ') {
          } else if (c == '/') {
          } else if (c == '(') {
            comment = COMMENT_TYPE_PARENTHESES;
          } else if (c == ';') {
            comment = COMMENT_TYPE_SEMICOLON;

          }     else     if     (char_counter     >=
(LINE_BUFFER_SIZE-1)) {
            report_status_message(STATUS_OVERFLOW);
            comment = COMMENT_NONE;
            char_counter = 0;
          } else if (c >= 'a' && c <= 'z') {
            line[char_counter++] = c-'a'+'A';
          } else {
            line[char_counter++] = c;
          }
        }
      }
    }
    protocol_auto_cycle_start();

    protocol_execute_realtime();
    if (sys.abort) { return; }
  }
```

Параметры при первоначальной настройки заносятся в энергонезависимую память EEPROM.

EEPROM – электрически стираемая программируемая память только для чтения. Гарантирует сохранность информации после последнего отключение питания. Необходимо помнить, что возможность перезаписи на устройство ограничено и составляет не более 100 000 раз. Поэтому следует аккуратно и внимательно относиться к вносимым данным. [17]

В файле montion_control.c следят за состоянием буфера в реальном времени и производят расчеты в принятом для Arduino UNO формате. Например, расчет вращения вектора по матрице преобразования. В листинге 7.7 представлен фрагмент кода с расчетом кривой по формуле, где r_T – поворотный вектор, phi – угол поворота, r – исходный вектор.

Для генерации дуги центром окружности является ось вращения, а радиус-вектор определяется от центра окружности до начального положения.

Листинг 7.7 – Пример расчета кривой по формуле

```
r_T = [cos(phi) -sin(phi);
                 sin(phi)  cos(phi] * r ;
```

Разберем на примере G-код. В листинге 7.8 представлен пример G-кода. [18]

Листинг 7.8 – Пример G-кода

```
G0 F00900
G1 F00300
M3
G0 x34.063 y19.75 z10
G1 x10 y19.75 z0
M5
```

G0 – Ускорение перемещение инструмента (холостой ход).
F00900 – Скорость рабочей подачи (мм/мин).
G1 – Координированное движение по осям X, Y, Z.
M3 – Начало вращение шпинделя. Команда включения.
X – Координата точки траектории по оси X.
Y – Координата точки траектории по оси Y.
Z – Координата точки траектории по оси Y. В нашем случае опускать ручку на 10 мм или опускать.
M5 – Команда выключения. Остановить вращение шпинделя.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была разработана и собрана модель планшетного перьевого графопостроителя с использованием Arduino UNO которая обеспечивает автоматическое рисование рисунков, чертежей и другой графической информации на бумагу формата А4.

В дальнейшем планируется доработать графопостроитель. Будет изменен каркас, чтобы был более устойчивым, вместе со специальным столом. Так же планируется добавить работу с внешним накопителем.

ПРИЛОЖЕНИЕ А

(справочное)
Библиография

1. Виды плоттеров: их отличия и характеристики, особенности выбора // URL: https://tehnopanorama.ru/orgtehnika/vidy-plotterov.html#i-3 (Дата обращения: 24.02.2020)
2. Рогожников Г.С. Прикладные физические технологии и компьютерные методы исследований: автоматизированные системы управления оптическими и оптико-механическими устройствами. Пособие для высших учебных заведений. – Саров, 2016. – 69 стр. (Дата обращения: 29.02.2020)
3. Шаговый двигатель NEMA17 JK42HS40 // URL: https://arduino.ua/prod2962-shagovii-dvigatel-nema17-jk42hs40-1704-13a-b (Дата обращения: 05.03.2020)
4. Шаговый 4-х фазный двигатель 28BYJ-48 с платой управления ULN2003 // URL: http://www.zi-zi.ru/docs/modules/info_28BYJ-48-5V_ULN2003.pdf (Дата обращения: 20.03.2020)
5. Драйвер шагового двигателя, A4988 // URL: https://iarduino.ru/shop/Expansion-payments/drayver-shagovogo-dvigatelya-a4988.html (Дата обращения: 09.03.2020)
6. Обзор драйвера шагового двигателя A4988 // URL: https://robotchip.ru/obzor-drayvera-shagovogo-dvigatelya-a4988/ (Дата обращения: 10.03.2020)
7. Драйвер шагового двигателя A4988 // URL: http://www.progdron.com/arduino-shield/arduino-shield/402-drajver-shagovogo- (Дата обращения: 10.03.2020)
8. DMOS Microstepping Driver with Translator And Overcurrent Protection. – Worcester, Massachusetts, 2009-2014. – 20 стр. (Дата обращения: 20.02.2020)
9. Сборка и настройка Arduino Uno и CNC Shield v.3 // URL: http://cnc-design.ru/sborka-arduino-uno-i-cnc-sheild-v3.html (Дата обращения: 01.04.2020)
10. Плата для ЧПУ CNC станка под Arduino UNO, CNC shield v3 и драйвера A4988 \ DRV8825 сборка, подключение, распиновка // URL: http://www.electronica52.in.ua/stanki-cnc--lazernye-i-drugie/plata-dlya-chpu-cnc-stanka (Дата обращения: 15.04.2020)
11. Сервопривод MG90S // URL: https://compacttool.ru/viewtovar.php?id=701 (Дата обращения: 1.05.2020)
12. Сервоприводы Ардуино SG90, MG995, MG996: схема подключения и управление // URL: https://arduinomaster.ru/motor-dvigatel-privod/servoprivody-arduino-sg90-mg995-shema-podklyuchenie-upravlenie/ (Дата обращения: 20.04.2020)
13. Первые шаги: Arduino IDE // URL: https://robotclass.ru/tutorials/arduino-ide/ (Дата обращения: 21.04.2020)
14. Что такое G-код для станков с ЧПУ // URL: https://vseochpu.ru/g-kody-dlya-chpu/ (Дата обращения: 27.04.2020)
15. Universal G-Code Sender Basics // URL: https://jtechphotonics.com/?p=4910 (Дата обращения: 04.05.2020)
16. Настройка GRBL – системные команды // URL: http://cnc-design.ru/proshivka-grbl-nastroika-sistemnyh-parametrov.html (Дата обращения: 05.05.2020)

17. Arduino EEPROM энергонезависимая память // URL: https://arduinoplus.ru/arduino-eeprom/ (Дата обращения: 07.05.2020)
18. Справочник G-код // URL: https://www.dreambird.ru/useful/definitions/g-code/ (Дата обращения: 11.05.2020)

ПРИЛОЖЕНИЕ Б

(рекомендуемое)
Наиболее употребляемые текстовые сокращения

ШИМ – Широтно-импульсная модуляция

ПО – Программное обеспечение

ЧПУ – Числовое программное управление

USB – Universal Serial Bus – Универсальная последовательная шина

Бод – Единица измерения скорости передачи данных по последовательному интерфейсу в битах в секунду

ПК – Персональный компьютер

ШД – Шаговый двигатель

COM порт – это стандартный двунаправленный последовательный порт компьютера

EEPROM – Electrically Erasable Programmable Read-Only Memory – электрически стираемое перепрограммируемое ПЗУ

IJK – Единичные векторы в прямоугольной системе координат

AC/DC – Alternating Current / Direct Current – Переменный ток, постоянный ток

ОЗУ – Оперативное запоминающее устройство

SRAM – Static Random Access Memory – Статическая память с произвольным доступом

Flash-память – Одна из разновидностей EEPROM ПЗУ

RX – Receive – Чтение данных

TX – Trancieve – Передача данных

LED – Light-emitting diode – Светодиод

SPI – Serial Peripheral Interface – Последовательный периферийный интерфейс

MOSI –Master Out Slave In – Служит для передачи данных от ведущего устройства ведомому

MISO – Master In Slave Out – Служит для передачи данных от ведомого устройства ведущему

SCK – Serial Clock – Служит для передачи тактового сигнала для ведомых устройств

SDA – Serial Data – Шина данных

SCL – Serial Clock – Тактовая шина

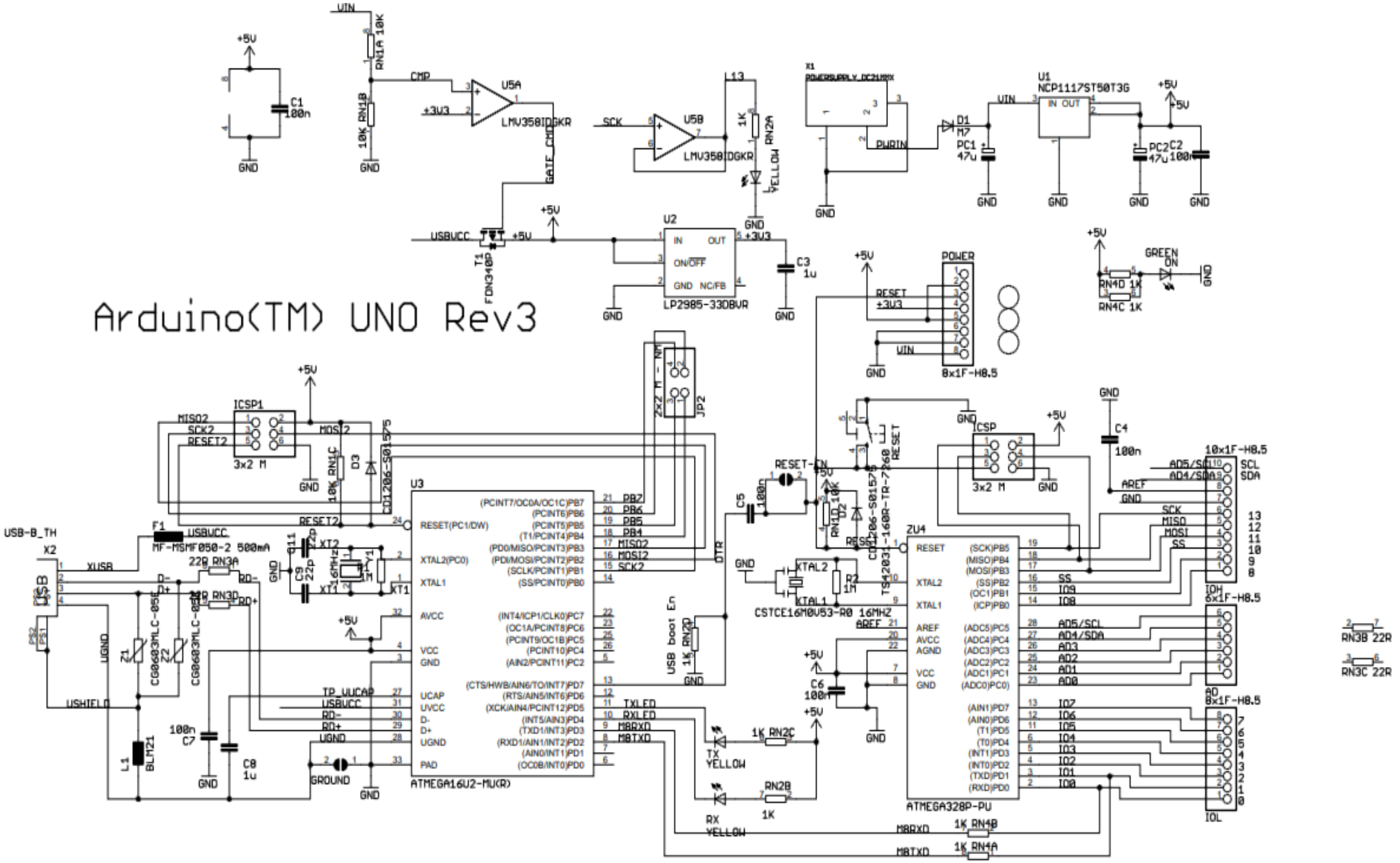
TWI – Two-wire Serial Interface– Двухпроводная шина, которая состоит из линии тактового сигнала (SCL) и линии данных (SDA)

UART – Universal Asynchronous Receiver/Transmitter – Узел вычислительных устройств, предназначенный для организации связи с другими цифровыми устройствами

I2C – Inter-Integrated Circuit – Последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов

ПРИЛОЖЕНИЕ В

Принципиальная схема Arduino UNO



ПРИЛОЖЕНИЕ Г

Принципиальная схема графопостроителя