

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  
" \_\_\_\_ " \_\_\_\_\_ 2023 г.

**БАКАЛАВРСКАЯ РАБОТА**

090301 Информатика и вычислительная техника

Система управления станка с ЧПУ на микроконтроллерной основе

Руководитель	_____	_____	доц. каф. ВТ, канд. техн. наук	В.Г. Середкин
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		В.С. Трошин
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____		В.Г. Середкин
	<i>подпись</i>	<i>дата</i>		

Красноярск 2023

## СОДЕРЖАНИЕ

Введение .....	3
1 Анализ предметной области .....	4
1.1 Анализ задания и обоснование выбора принимаемого решения .....	4
1.2 Аналогии СЧПУ .....	5
2 Разработка системы управления трехосевым станком с ЧПУ .....	10
2.1 Структурная информационная модель процесса управления СЧПУ и схема электрическая структурная на ее основе .....	10
2.2 Обоснование выбора функциональных узлов и разработка схемы электрической функциональной системы управления СЧПУ .....	13
2.2.1 Выбор микроконтроллерной платформы .....	13
2.2.2 Выбор драйверов для шаговых двигателей .....	17
2.2.3 Выбор шаговых двигателей .....	21
2.3 Конструктивные изменения станка для работы в трехкоординатном варианте .....	26
3 Моделирование функциональных узлов разрабатываемой системы управления на натуральных макетах .....	28
3.1 Обоснование выбора аппаратных и программных средств для моделирования функциональных узлов .....	28
3.2 Примеры натурального моделирования типовых узлов разрабатываемой системы .....	30
3.3 Разработка схемы электрической принципиальной прототипа модуля управления станком с ЧПУ .....	33
4 Разработка ПО системы управления СЧПУ на основе промышленного контроллера .....	35
4.1 Разработка алгоритма функционирования системы управления СЧПУ	35
4.2 Обоснование выбора и необходимости разработки ПО .....	37
4.3 Пример программного кода .....	41
Заключение .....	44
Список используемых источников .....	45
ПРИЛОЖЕНИЕ А .....	46
ПРИЛОЖЕНИЕ Б .....	47
ПРИЛОЖЕНИЕ В .....	48
ПРИЛОЖЕНИЕ Г .....	54

## ВВЕДЕНИЕ

Тема данной выпускной работы актуальная, так как заявленная тематика работы обеспечивает разработку и использование аппаратно-программных средств как основы систем управления электромеханическими агрегатами, такими как станки с числовым программным управлением, на микроконтроллерном уровне.

Целью данной ВКР является разработка системы управления сверлильно-фрезерным трехосевым станком с числовым программным управлением.

Для достижения данной цели были поставлены следующие задачи:

- проанализировать задание на ВКР и обосновать выбор принимаемого решения;
- разработать систему управления трехосевым станком с ЧПУ;
- провести моделирование функциональных узлов разрабатываемой системы управления на натуральных макетах;
- разработать ПО системы управления СЧПУ на основе промышленного контроллера.

## **1 Анализ предметной области**

### **1.1 Анализ задания и обоснование выбора принимаемого решения**

В современном мире автоматизация производственных процессов становится все более актуальной. Одним из наиболее важных направлений в данной области является разработка систем управления станками с ЧПУ. В этой связи, разработка такой системы является актуальной задачей, которую необходимо решить.

Целью выпускной квалификационной работы (ВКР) является разработка системы управления сверлильно-фрезерным станком с числовым программным управлением.

## 1.2 Аналоги СЧПУ

На сегодняшний день существует множество различного программного и аппаратного обеспечения различного применения и различной производительности и функциональности. Однако большим спросом являются готовые станки с ЧПУ (числовое программное управление). Как пример бюджетного решения станок «CNC3018Pro» из Китая от компании ККМООН (Рисунок 1).



Рисунок 1 – Станок «CNC3018Pro»

Ниже представлены его характеристики (Таблица 1):

Таблица 1 – Характеристики станка CNC3018Pro

Характеристика	Значение
Программное обеспечение управления	GRBL
Рабочая зона	300*180x40 мм
Размер оправы	420*355*280 мм
Шпиндель	775 мотор шпинделя (12-36 в) 24 В: 7000об/мин, 36 В: 9000об/мин
Патрон шпинделя	ER11
Шаговый двигатель	1.3А
Источник питания	24 В/5,6 А (110 В - 240 В)
Поддерживаемая система	Windows XP / Win 7 / Win 8 / Win 10.
Гравировальные материалы	пластик, дерево, акрил, ПВХ, печатная плата и другие мягкие материалы.
Поддерживаемые форматы	jpg / bmp / png
Порт связи	Micro USB
Открытость ПО	нет
Возможность доработки	заводские компоненты
Цена	88000 рублей

Так же на рынке широко представлены варианты от Российского производителя, например, станок «ССТ/МТС» от компании Bungard (Рисунок 2).



Рисунок 2 – Станок «ССТ/МТС»

Характеристики данного станка (Таблица 2):

Таблица 2 – Характеристики станка CCD/МТС

Характеристика	Значение
Шпиндель	КАVO 150 Вт, 25 000-60 000 об/мин
Обрабатываемая поверхность	325 × 495 мм. Перемещение по оси Z до 33 мм
Программное обеспечение принимает форматы	HP/GL (фрезерование), Exellon (сверление)
Дискретность шага	0,0254 мм
Скорость фрезерования	до 93 мм/сек (5,6 м/мин)
Скорость сверления	до 5 проходов/сек (18000 отверстий/час)
Напряжение питания	230 В, 50 Гц
Габаритные размеры (Д × Ш × В)	500 × 600 × 450 мм
Открытость ПО	нет
Возможность доработки	любые подходящие компоненты
Цена	20000 рублей



Вывод:

Были рассмотрены разные готовые системы с ЧПУ. Представленные модели имеют множество преимуществ, но также есть некоторые недостатки, в частности довольно большая цена, закрытость программного управления и обеспечения, зачастую отсутствие возможности улучшения и аппаратной поддержки со стороны производителя.

В связи с чем было принято решение разработать свой программно-аппаратный комплекс на основе заводского прототипа отечественного производства. В качестве исполнительного устройства послужил двухосевой электромеханический сверлильно-фрезерный станок, производимый промышленностью СССР для учебных и лабораторных целей, представленный на рисунке 3.



Рисунок 3 – Прототип отечественного электромеханического двухосевого агрегата

## 2 Разработка системы управления трехосевым станком с ЧПУ

### 2.1 Структурная информационная модель процесса управления СЧПУ и схема электрическая структурная на ее основе

Структурная информационная модель процесса управления СЧПУ представляет собой совокупность блоков объекта управления и системы управления станком с ЧПУ (Рисунок 4).

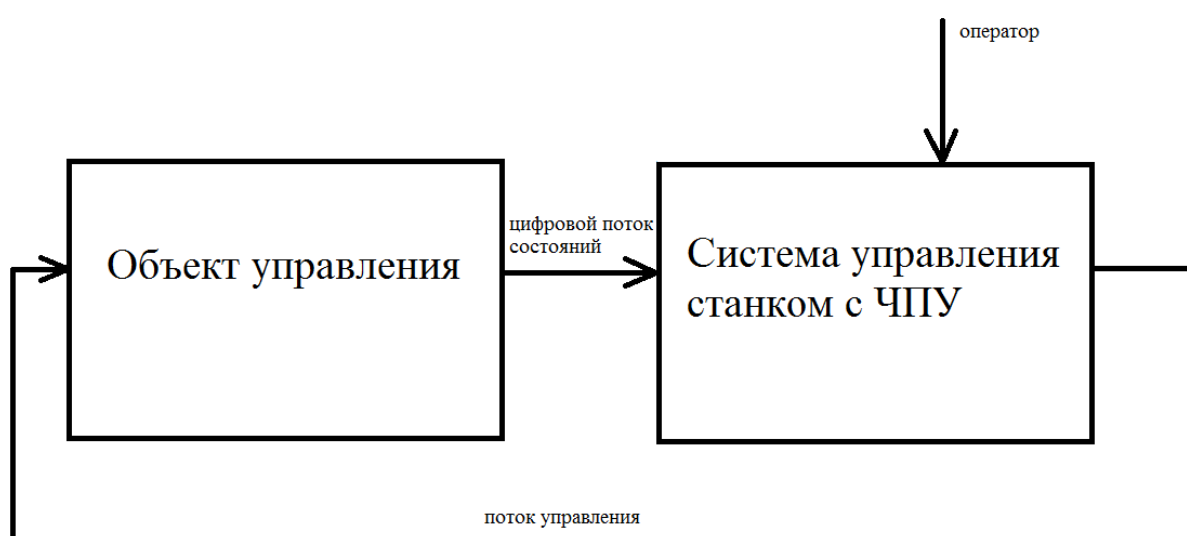


Рисунок 4 – Структурная информационная модель процесса управления СЧПУ

Объект управления порождает цифровой поток данных от 6 концевых датчиков (по два для каждой из осей X, Y и Z). Поток управления обеспечивает возможность управляющего воздействия на три шаговых двигателя по оси X, Y и Z. Оператор имеет возможность взаимодействия с системой управления СЧПУ посредством стандартного набора периферийных устройств ПК.

На основе представленной информационной модели была разработана структура станка с ЧПУ с интегрированным контроллером представленная на рисунке 5.

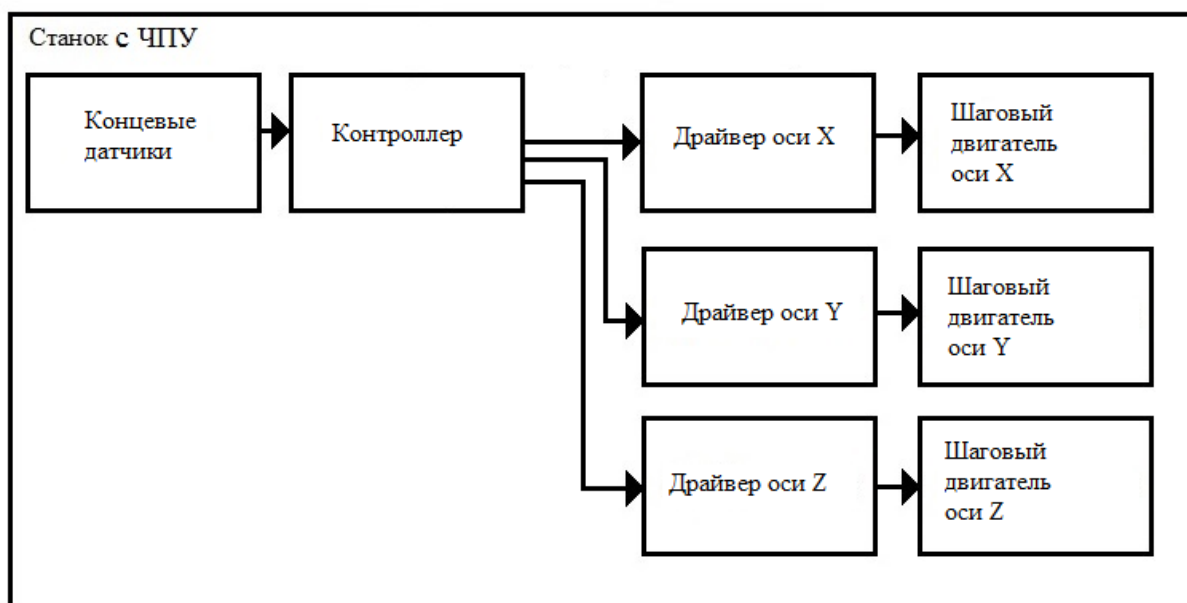


Рисунок 5 – Структура станка с ЧПУ с интегрированным контроллером

Предложенная структура включает набор концевых датчиков, связанных с портами контроллера, набор драйверов (формирователей), связанных с выходными портами контроллера и шинами управления шаговыми двигателями.

Структура разрабатываемой системы управления СЧПУ представлена на рисунке 6.

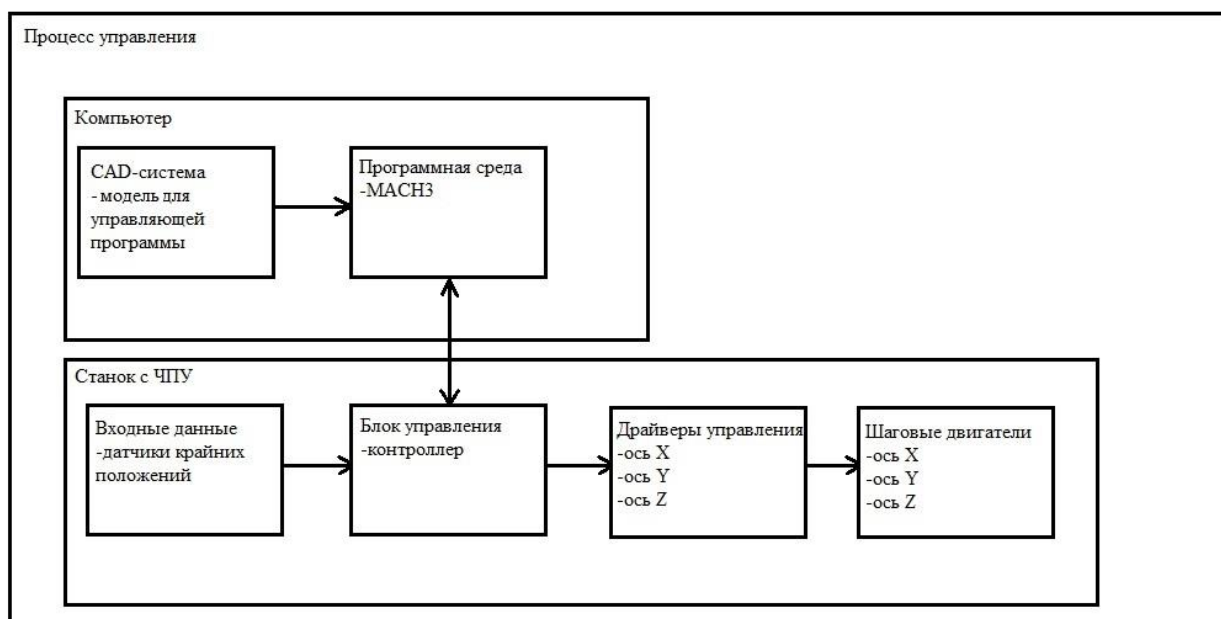


Рисунок 6 – Структура разрабатываемой системы управления СЧПУ

Система управления разбивается на два уровня: верхний уровень, состоящий из стандартного ПК на системе Windows не ниже 7 версии с установленным специализированным программным обеспечением, и связанный по интерфейсу USB с микроконтроллером интегрированный в состав станка с ЧПУ и нижний уровень, реализованный на программируемом контроллере и обеспечивающий функции управления шаговыми двигателями по осям X, Y и Z.

Программное обеспечение верхнего уровня представлено в виде комплекса программных продуктов по разработке модели для печати, в дальнейшем преобразованную в управляющую программу (УП) основанную на G-CODE, и программы, использующей эти коды для управления шаговыми двигателями станка.

Программные продукты для проектирования и создания УП можно устанавливать и использовать на любом ПК. Программа управления станком должна быть установлена на компьютере, непосредственно подключенном к станку с ЧПУ.

## 2.2 Обоснование выбора функциональных узлов и разработка схемы электрической функциональной системы управления СЧПУ

### 2.2.1 Выбор микроконтроллерной платформы

Для обоснования выбора микроконтроллерной платформы были рассмотрены несколько наиболее популярных моделей контроллеров, таких как Buildbotics, Masso, DrufelCNC, Arduino Uno. Все эти контроллеры представлены на рисунке 7.



Рисунок 7 – Рассматриваемые контроллеры

Контроллер станка с ЧПУ Buildbotics — это контроллер ЧПУ с открытым исходным кодом, который используется в таких ЧПУ, как фрезерный станок с ЧПУ Onefinity. Контроллер имеет Raspberry Pi 3, подключенный к микроконтроллеру ATmega192a3u. В этом контроллере G-код, сгенерированный на компьютере, отправляется в Raspberry Pi, который отправляет его в микроконтроллер для обработки и передачи сигналов в драйверы шагового двигателя и другие компоненты. Цена данного контроллера составляет около 50000 рублей плюс доставка. Хотя популярность контроллера Buildbotics только начинает расти, у него есть управляемый компанией форум, через который можно получить ответы на свои вопросы.

Masso G3 имеет систему управления в 3-осевом, 4-осевом и 5-осевом вариантах. Так же Masso G3 имеет управляющее программное обеспечение для трех типов станков с ЧПУ, фрезерных станков с ЧПУ, токарных станков с ЧПУ и станков лазерной резки с ЧПУ. Плата контроллера имеет довольно обширные варианты ввода / вывода с 24 оптически изолированными входами и 18 выходами. Что касается подключения, Masso G3 может принимать G-код через WiFi с любого устройства (компьютера, телефона) через свое программное обеспечение MASSO Link. Цена данного контроллера варьируется от 60000 до 75000 рублей. У Masso G3 есть специализированный онлайн-форум, а также опция поддержки по электронной почте и портал поддержки, где можно оставить заявку в случае возникновения проблем.

Arduino Uno - это плата микроконтроллера с открытым исходным кодом, основанная на микроконтроллере Microchip ATmega328P и разработанная Arduino.cc и первоначально выпущена в 2010 году. Плата оснащена наборами контактов цифрового и аналогового ввода/вывода (I/O), которые могут быть подключены к различным платам расширения. Плата имеет 14 цифровых выводов ввода-вывода, 6 аналоговых выводов ввода-вывода и программируется с помощью Arduino IDE (интегрированная среда разработки) через USB-кабель. Он может питаться от USB-кабеля или от внешней 9-вольтового источника питания напряжением от 7 до 20 вольт. И хотя он уступает другим специализированным контроллерам по некоторым параметрам, таким как точность позиционирования, необходимость подготовки и отладки как отдельных компонентов так и всей системы, он имеет ряд достоинств, как например возможность самому написать весь управляющий код, что поможет лучше понимать механику работы системы, гигантский функционал и все направленность, а также небольшая цена относительно других контроллеров – от 500 до 2000 рублей, что делает его лучшим решением при разработке прототипа системы управления станком.

После анализа вышеперечисленных контроллеров, для итогового решения системы был выбран контроллер модели «BSMCE04U» фирмы «DrufelCNC». Этот контроллер обладает хорошими показателями, как в части управления двигателями, так и в количестве входных сигналов, что обеспечивает возможность модернизации станка в будущем. В то же время он один из самых простых контроллеров среди аналогов. Это означает, что для получения необходимых настроек требуются минимальные навыки, что также дает свободу использовать различные варианты шаговых двигателей, драйверов, если нужно выполнить замену функциональных узлов, не меняя плату управления (Рисунок 8).

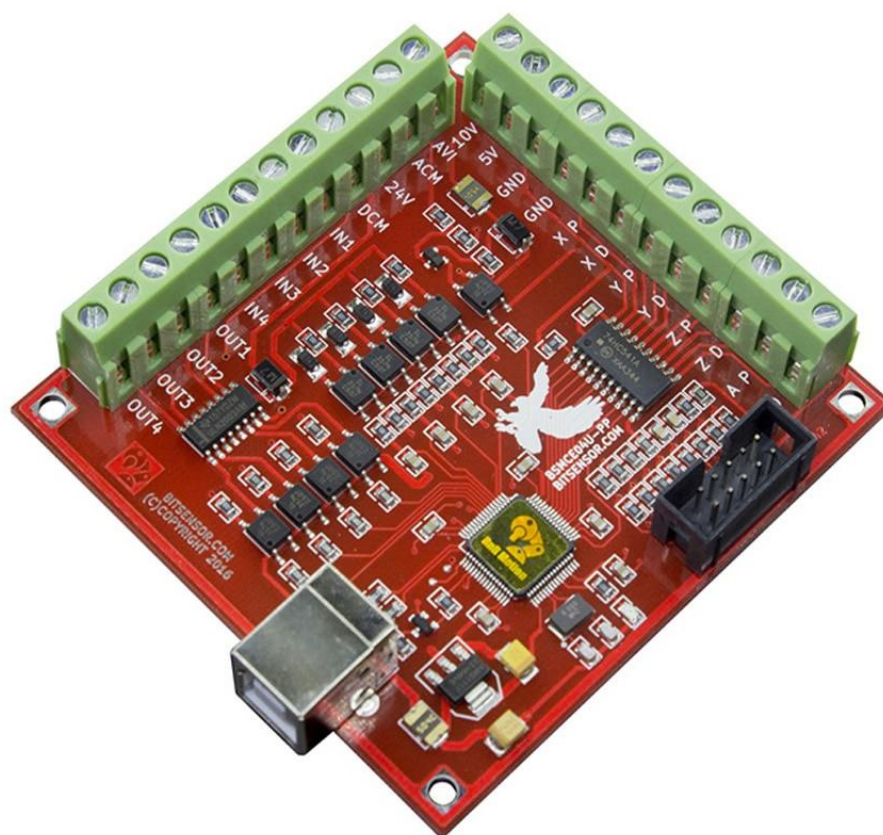


Рисунок 8 – Контроллер BSMCE04U

Его характеристики представлены ниже в таблице 3.

Таблица 3 – Характеристика контроллера BSMCE04U

Характеристика	Значение
Напряжение питания, В	12-24
Входов состояния, шт	4
Выходов состояния, шт	4
Информационные выходы на двигатели	По два на оси X, Y, Z, A
Связь с ПК	Шина USB
Цена, руб	2000



## 2.2.2 Выбор драйверов для шаговых двигателей

При выборе драйвера шагового двигателя следует отталкиваться не только от характеристик двигателя, но и от необходимой точности выполнения перемещения. Так же следует оставить возможность для модификации системы в случае модернизации станка или замены вышедших из строя двигателей на более современные и надежные.

После анализа рынка драйверов для шаговых двигателей было выбрано 3 наиболее подходящих драйвера для данного мотора: CW-5045, ТВ6560, М542-05 (Рисунок 9).



Рисунок 9 – Рассматриваемые драйвера

Таблица 4 – Характеристики драйвера CW – 5045

Характеристика	Значение
Напряжение питания, В	24 – 50
Номинальный ток, А	1,3 – 4,5
Микрошаг	1 – 1/256
Понижения тока при простое	50%
Защита от короткого замыкания	присутствует
Защита от перегрева	присутствует

Таблица 5 – Характеристики драйвера ТВ 6560

Характеристика	Значение
Напряжение питания, В	10 – 36
Номинальный ток, А	1 – 3
Микрошаг	1 – 1/16
Понижения тока при простое	50%
Защита от короткого замыкания	отсутствует
Защита от перегрева	отсутствует

Таблица 6 – Характеристики драйвера M542-05

Характеристика	Значение
Напряжение питания, В	20 – 50
Номинальный ток, А	1,2 – 5,1
Микрошаг	1 – 1/256
Понижения тока при простое	50%
Защита от короткого замыкания	присутствует
Защита от перегрева	присутствует

Согласно данным выше характеристикам (Таблица 4, таблица 5, таблица 6), был выбран драйвер M542-05. Данный драйвер удовлетворяет не только параметрам шагового двигателя и условиям необходимой точности, но также достаточно универсален для дальнейшей модернизации станка, установки более современных и производительных двигателей. Этот драйвер так же является лидером среди конкурентов в соотношении цена/функциональность, что так же является плюсом при выборе данного драйвера (Рисунок 10).



Рисунок 10 – Драйвер M542-05

### 2.2.3 Выбор шаговых двигателей

В качестве двигателей осей X и Y были использованы штатные шаговые двигатели ДШИ200 (Рисунок 11). Их характеристики представлены ниже в таблице 7.

Таблица 7 – Характеристики шагового двигателя ДШИ200

Характеристика	Значение
Напряжение питания, В	30 ( $\pm 2$ )
Номинальный ток, А	1,5 ( $\pm 0,1$ )
Максимальный ток, А	2
Максимальная чистота работы, кГц	1
Единичный шаг, градус	1,8
Число обмоток	4
Максимальная потребляемая мощность, Вт	16,7



Рисунок 11 – Шаговый двигатель ДШИ200

В качестве двигателя оси Z был использован шаговый двигатель Nema 17 (Рисунок 12). Этот двигатель имеет меньшие габариты относительно других двигателей серии, таких как Nema 23 или Nema 32, но при этом не сильно уступает им в мощности или точности. Его характеристики представлены ниже в таблице 8.

Таблица 8 – Характеристики шагового двигателя Nema 17

Характеристика	Значение
Напряжение питания, В	3,6 на фазу
Номинальный ток, А	1,5 на фазу
Число фаз	2
Единичный шаг, градус	1,8
Число обмоток	4



Рисунок 12 – Шаговый двигатель Nema 17

После обоснования выбора функциональных узлов была разработана функциональная схема системы управления станком с ЧПУ (Рисунок 13). Датчики крайних положений НОМЕ на схеме представлены ключами. Контроллер системы управления представлен на схеме как CPU и является основным элементом управления. К нему подключены все датчики положения и драйвера. Так же к контроллеру подключается компьютер с необходимым программным обеспечением через интерфейс USB.

Шаговые двигатели, представленные на схеме как Step Motor, управляются драйверами Driver X, Driver Y и Driver Z. У каждого драйвера есть свое питание в 30В. Управляющие сигналы генерирует контроллер в соответствии с выполняемой командой. Вход DIR отвечает за направление вращения двигателя, выход STEP отвечает за угол вращения (каждый сигнал изменяет угол на 1,8 градуса).



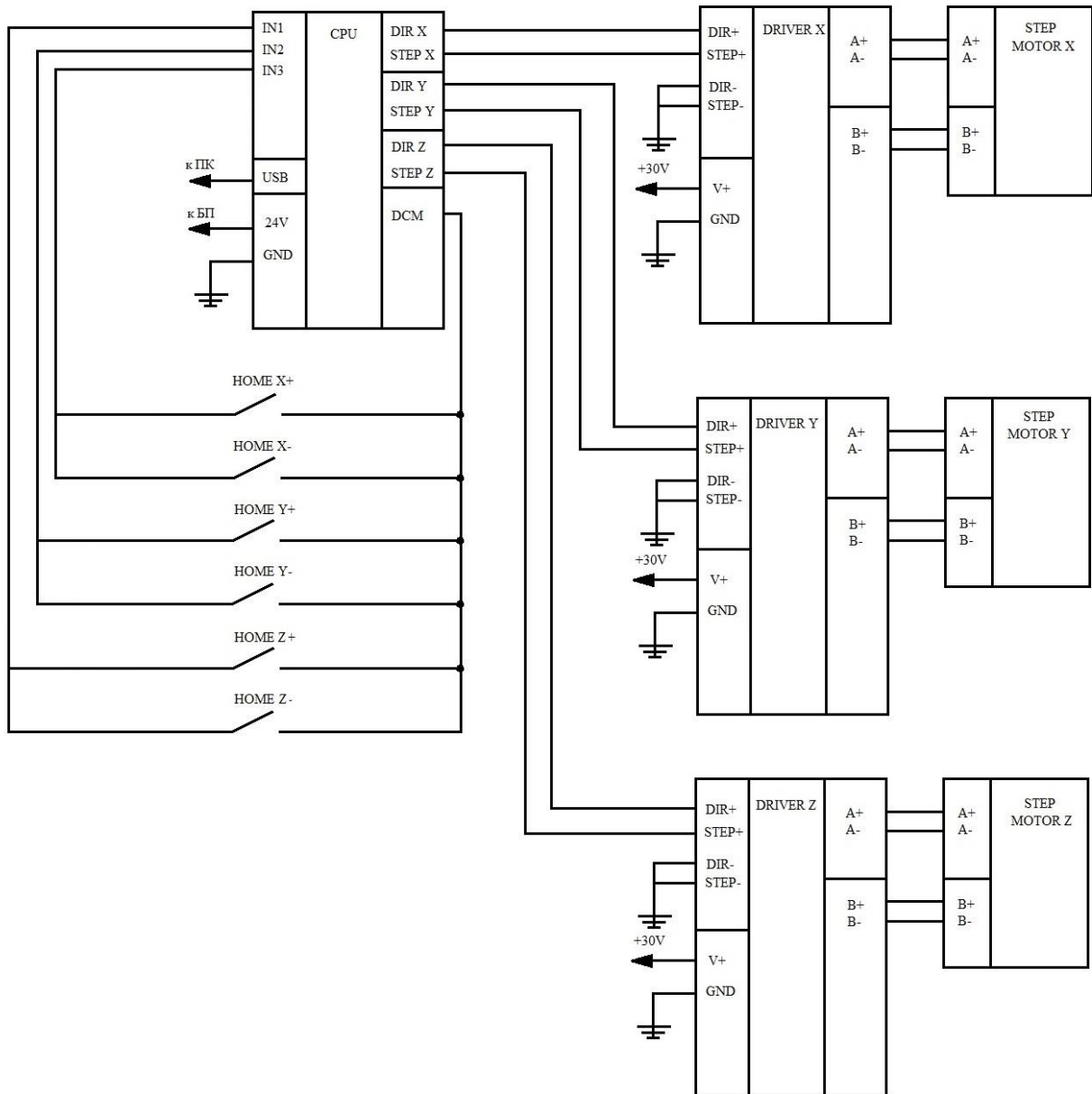


Рисунок 13 – Функциональная схема разрабатываемой системы управления

## **2.3 Конструктивные изменения станка для работы в трехкоординатном варианте**

В соответствии с заданием на ВКР станок имел две рабочие оси, требовалось внести конструктивные изменения для увеличения функционала путем добавления третьей оси.

Для этого были внесены следующие конструктивные изменения:

- изменено положение оси  $Y$  с горизонтально – ориентированного положения на вертикальное;

- увеличены боковые пластины путем добавления штанг, выточенных из стального прута, соединяющих основание подвижной платформы и боковые упоры для направляющих оси  $Y$ ;

- добавлен скользящий стол оси  $Z$ , состоящий из штатных направляющих, червячной передачи и каретки, к которому был установлен шаговый двигатель Nema 17;

- был изменен метод крепления шпинделя со штатного на более подходящий для работы после всех конструкционных изменений.

Готовый вид станка представлен на рисунке 14.

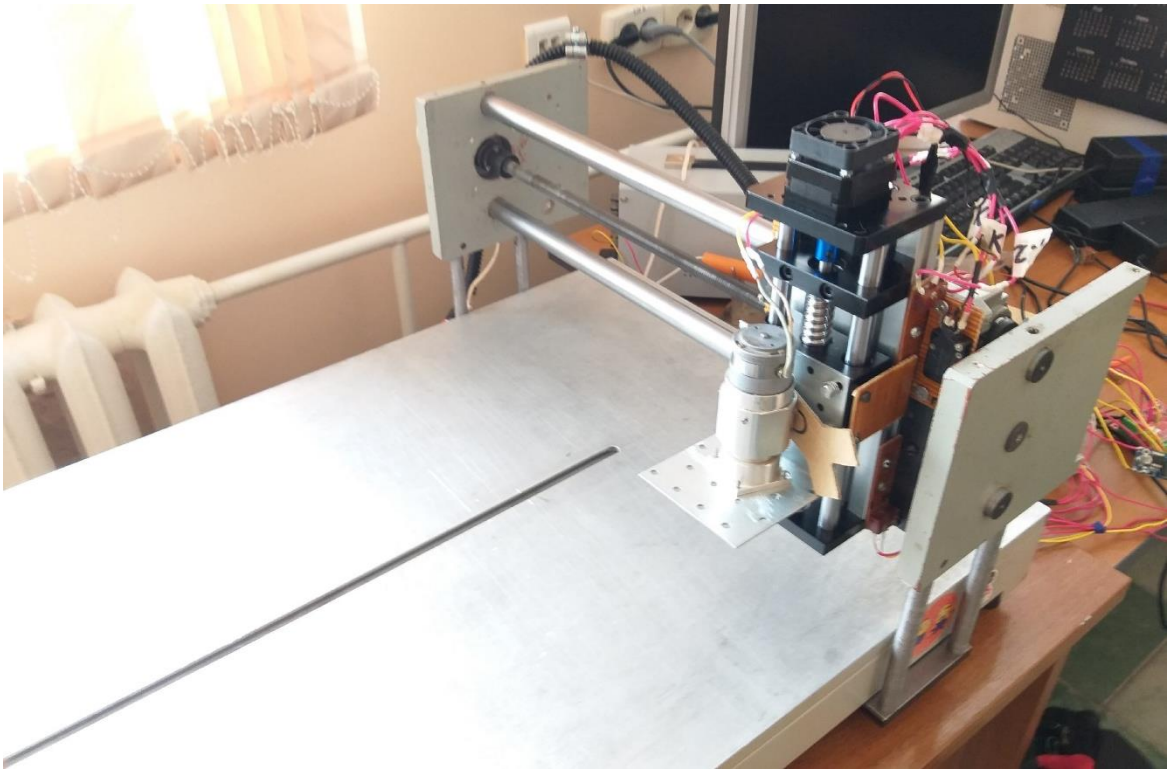


Рисунок 14 – Конструктивные изменения станка с ЧПУ

### 3 Моделирование функциональных узлов разрабатываемой системы управления на натуральных макетах

#### 3.1 Обоснование выбора аппаратных и программных средств для моделирования функциональных узлов

Из всех рассматриваемых контроллеров для прототипирования больше всего подходит контроллер Arduino Uno. Его характеристики представлена ниже в таблице 9.

Таблица 9 – Характеристики Arduino Uno

Характеристика	Значение
Микроконтроллер	ATmega328P
Ядро	8-битный AVR
Тактовая частота	16 МГц
Flash-память	32 КБ
РАМ-память	2 КБ
Пины с ШИМ	6
Разрядность ШИМ	8 бит
Напряжение логических уровней	5 В
Входное напряжение питания	через USB: 5 В
Максимальный выходной ток пина 3V3	150 мА
Максимальный выходной ток пина 5V	1 А

На рисунке 15 представлена схема электрическая принципиальная контроллера, которая использовалась при моделировании системы станка с ЧПУ.

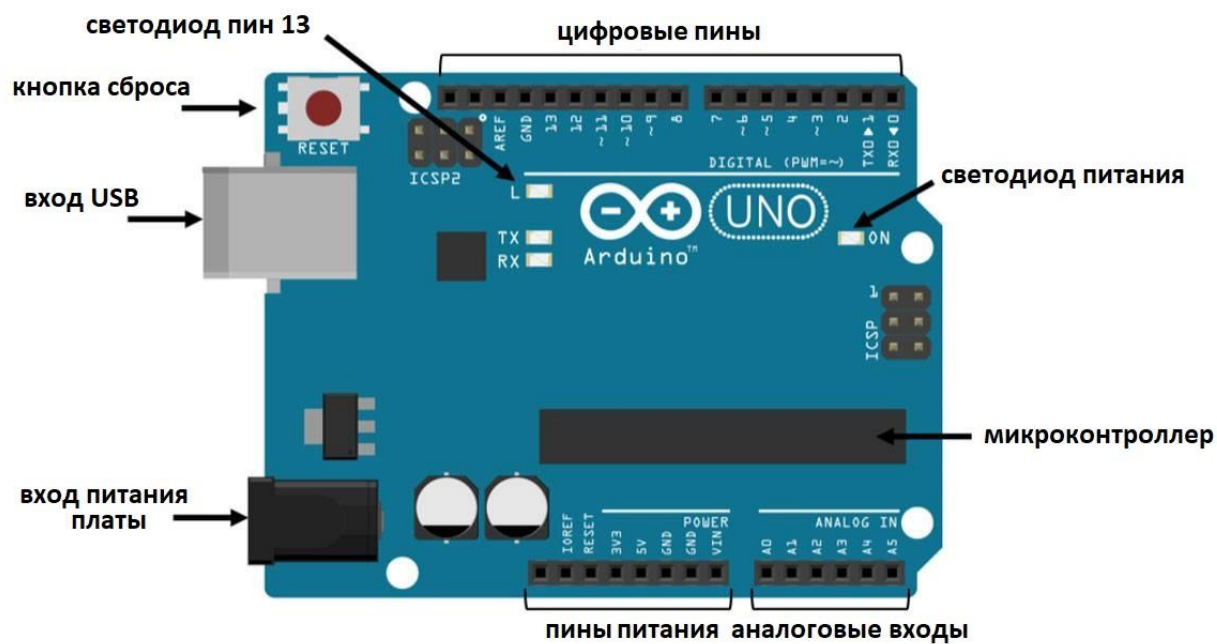


Рисунок 15 – Схема электрическая принципиальная контроллера прототипа

## 3.2 Примеры натурального моделирования типовых узлов разрабатываемой системы

Для работы прототипа была взята из открытого интернет ресурса и адаптирована код-программа за авторством AlexGyver (Приложение В).

Была использована библиотека GyverStepper v2 взятая с того же интернет ресурса (Рисунок 16).

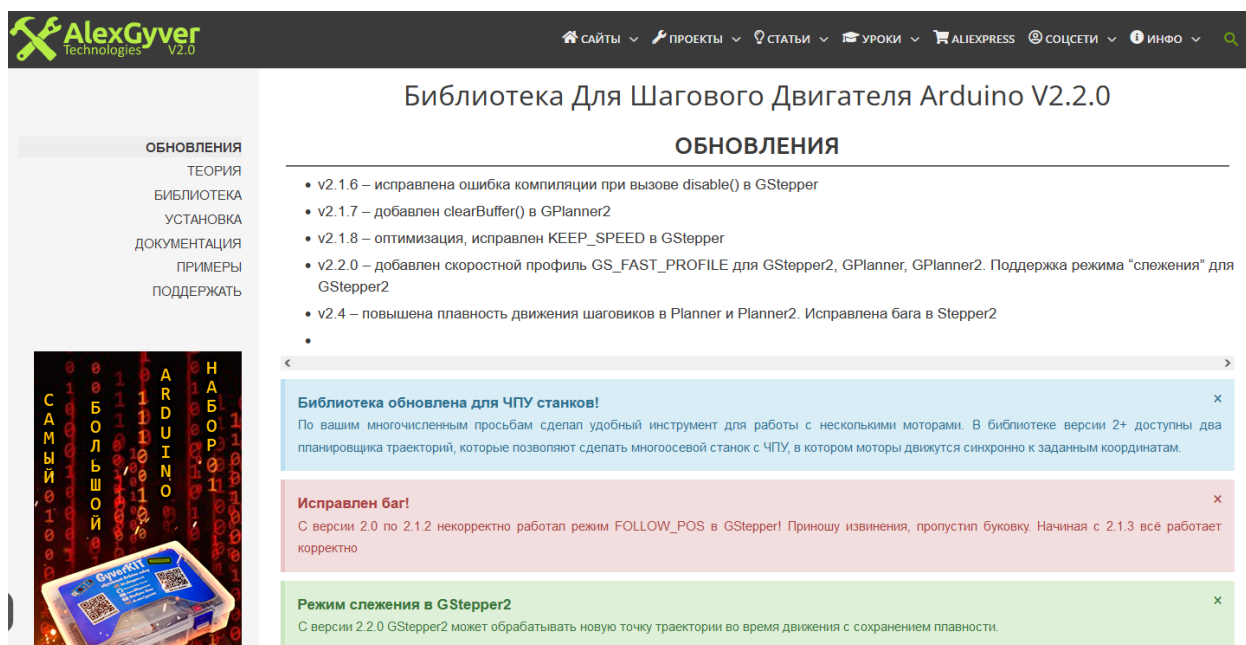


Рисунок 16 – Используемая библиотека

Библиотека содержит набор инструментов для разных сценариев работы с шаговыми моторами:

– `StepperCore.h` [класс `Stepper`]: ядро всех остальных классов, умеет быстро щелкать пинами (AVR) и делать один шаг для настроенного типа драйвера. Поддерживает 4 фазы шаг/полушаг, а также `step-dir` драйверы.

– `GyverStepper.h` [класс `GStepper`]: основная тяжелая библиотека, много настроек. Движение одного мотора с ускорением к заданной позиции или вращение с заданной скоростью. Не очень оптимальная работа в прерывании таймера.

– `GyverStepper2.h` [класс `GStepper2`]: новая облегченная версия `GyverStepper`, практически полностью с ней совместима. Более оптимальный целочисленный гибридный алгоритм движения с ускорением, легкий вес. Оптимизировано для работы в прерывании таймера.

– `GyverPlanner.h` [класс `GPlanner`]: многоосевой планировщик траектории, движение с ускорением (2 порядок). Остановка в каждой точке. Оптимальная работа в прерывании таймера.

– `GyverPlanner2.h` [класс `GPlanner2`]: многоосевой планировщик траектории, движение с ускорением (2 порядок). Планирование скорости на маршруте, оптимальное движение по точкам. Оптимальная работа в прерывании таймера.

Библиотека имеет два режима работы с мотором, устанавливается при помощи `setRunMode(mode)`, где `mode`:

– `FOLLOW_POS` - режим плавного движения к заданной позиции с ускорением и ограничением скорости.

– `KEEP_SPEED` - режим вращения с заданной скоростью (знак скорости задает направление вращения).

После подключения контроллера и адаптации кода была запущена тестовая программа, результат которой показан на рисунке 17 и рисунке 18.

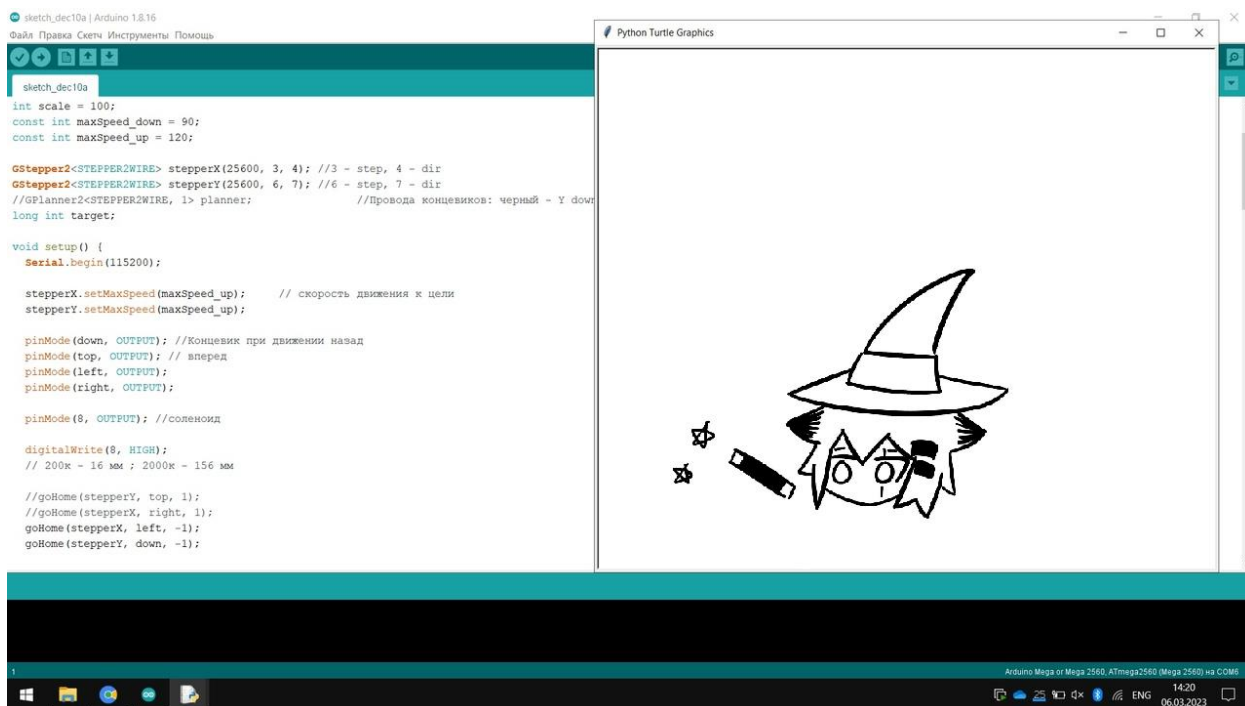


Рисунок 17 – Тестовый код программа на основе прототипа



Рисунок 18 – Результат выполнения работы



### 3.3 Разработка схемы электрической принципиальной прототипа модуля управления станком с ЧПУ

На основании схемы электрической функциональной была разработана схема электрическая принципиальная прототипа модуля управления, включающая компоненты, представленные на рисунке 19, рисунке 20 и рисунке 21.

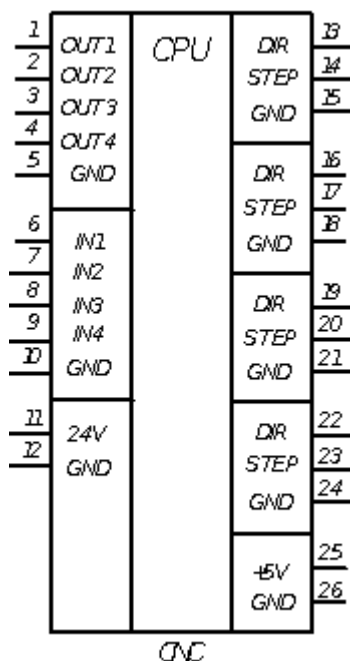


Рисунок 19 – Условное графическое обозначение контроллера системы управления

Контроллер имеет 4 выхода для подключения концевых выключателей (контакты 6-10), 4 блока DIR/STEP/GND для подключения до 4х осевых двигателей (контакты 13-15 для оси X, контакты 16-18 для оси Y, контакты 19-21 для оси Z, контакты 22-24 для оси A), а также вход питания USB 5В (контакт 25). Сам контроллер работает от шины 12-24В (контакт 11).

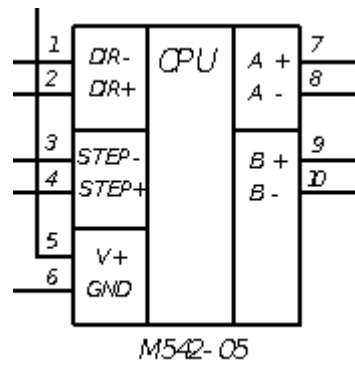


Рисунок 20 – Условное графическое обозначение драйвера шагового двигателя

Драйвер имеет 4 информационных входа для подачи сигнала от контроллера (контакты 1-4), вход питания (контакты 5-6), а также 4 выхода на шаговые двигатели (контакты 7-10).

ХР1

Конт.	Цель
A1	A
A2	B
A3	Б
A4	Г

Рисунок 21 – Условное графическое обозначение разъема шагового двигателя

Шаговый двигатель имеет 4 информационных входа для сигналов от драйвера (контакты A1-A4).

Схема электрическая принципиальная прототипа модуля управления, включающая разъемы, представлена в приложении Б.

## 4 Разработка ПО системы управления СЧПУ на основе промышленного контроллера

### 4.1 Разработка алгоритма функционирования системы управления СЧПУ

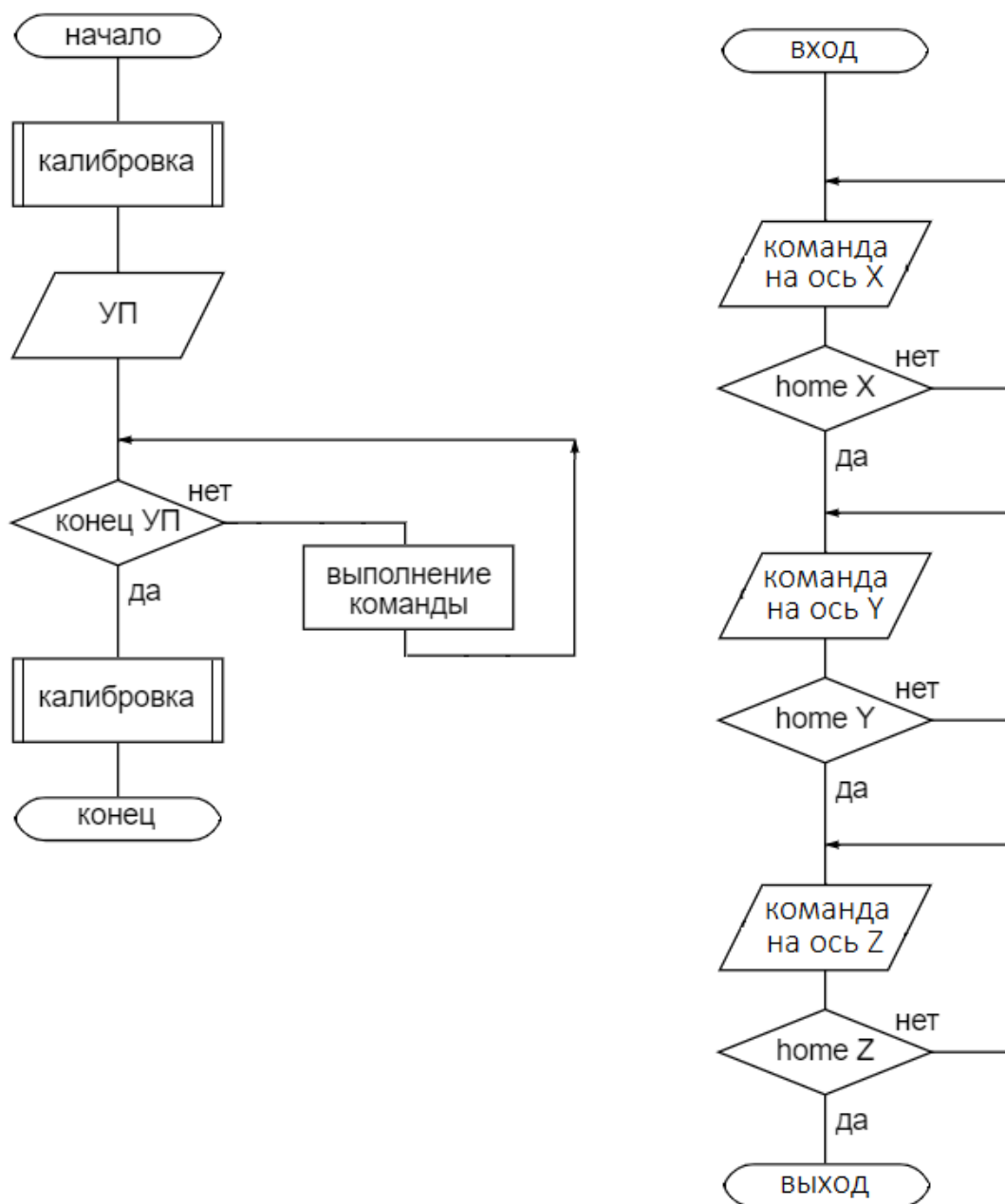


Рисунок 22 – Алгоритм функционирования системы

Описание общего алгоритма работы системы управления (Рисунок 22):

1) после подачи напряжения питания, происходит начальная калибровка, при успешном выполнении которой, станок переходит в состояние готовности;

2) в режиме готовности контроллер принимает управляющую команду (УК) и приступает к выполнению программы;

3) после выполнения текущей команды выполняется следующая команда;

4) если полученная команда завершения УП, то контроллер отключает шпиндель и выполнял автоматическую калибровку для перехода в режим готовности или отключения питания, иначе выполняется следующая команда.

Описание алгоритма функции калибровки:

1) в начале калибровки на драйвер двигателя оси X подаются сигналы для передвижения шпинделя в положение с координатой  $X = 0$ , постоянно опрашивая датчик положения оси X;

2) далее аналогичная операция проводится с осью Y и с осью Z, результатом которой станет положение шпинделя в начале координат;

3) окончание калибровки переход станка в состояния готовности.

Алгоритм работы системы управления был разработан для достижения максимальной эффективности и производительности системы.

Так же стоит отметить, что во время выполнения любой из операций отслеживаются состояния датчиков крайнего положения шпинделя и в случае если появится сигнал от одного из них, выполнение текущей и последующих команд приостановится и в компьютер будет отправлено сообщение о возникновении аварийной ситуации.

## 4.2 Обоснование выбора и необходимости разработки ПО

На рынке программного обеспечения для станков с ЧПУ так же, как и на рынке готовых станков существует большое количество продуктов, как дорогих и имеющих большой функционал и стабильность, так и дешевых, менее функциональных. Характеристики рассматриваемых ПО представлены ниже в таблице 10.

Таблица 10 – Характеристики рассматриваемых ПО

Характеристика	Aspire	ESam	HeeksCNC	Mach3
1. Лицензия	Проприетарная	Проприетарная	Проприетарная	Бесплатная,
2. Функциональность	Широкий набор функций для создания электронных схем и 3D моделей для ЧПУ	Широкий набор функций для создания электронных схем и 3D моделей для ЧПУ, симуляция и анализ	Широкий набор функций для создания электронных схем и 3D моделей для ЧПУ	Базовый набор функций для работы с электронными схемами и 3D моделями для ЧПУ
3. Платформа	Windows	Windows	Windows	Windows, Linux
4. Поддержка	Поддержка через систему обратной связи	Техническая поддержка, обучение и курсы обучения	Техническая поддержка через систему обратной связи	Онлайн форум, wiki и Github
5. Поддержка языков разработки	Verilog, VHDL	SystemC, Verilog-A/AMS, VHDL-AMS	Поддержка языков разработки C, Basic	C/C++, Python, Lua
7. Форматы, предназначенные для станка с ЧПУ	Gerber, Excellon N/C Drill, BOM	Sieb&Meyer, HPGL123, Gerber	ODB++, Gerber RS-274X / Gerber X2, Excellon N/C Drill	Excellon N/C Drill, Gerber, G-code

Их интерфейсы представлены ниже на рисунке 23, рисунке 24 и рисунке 25.

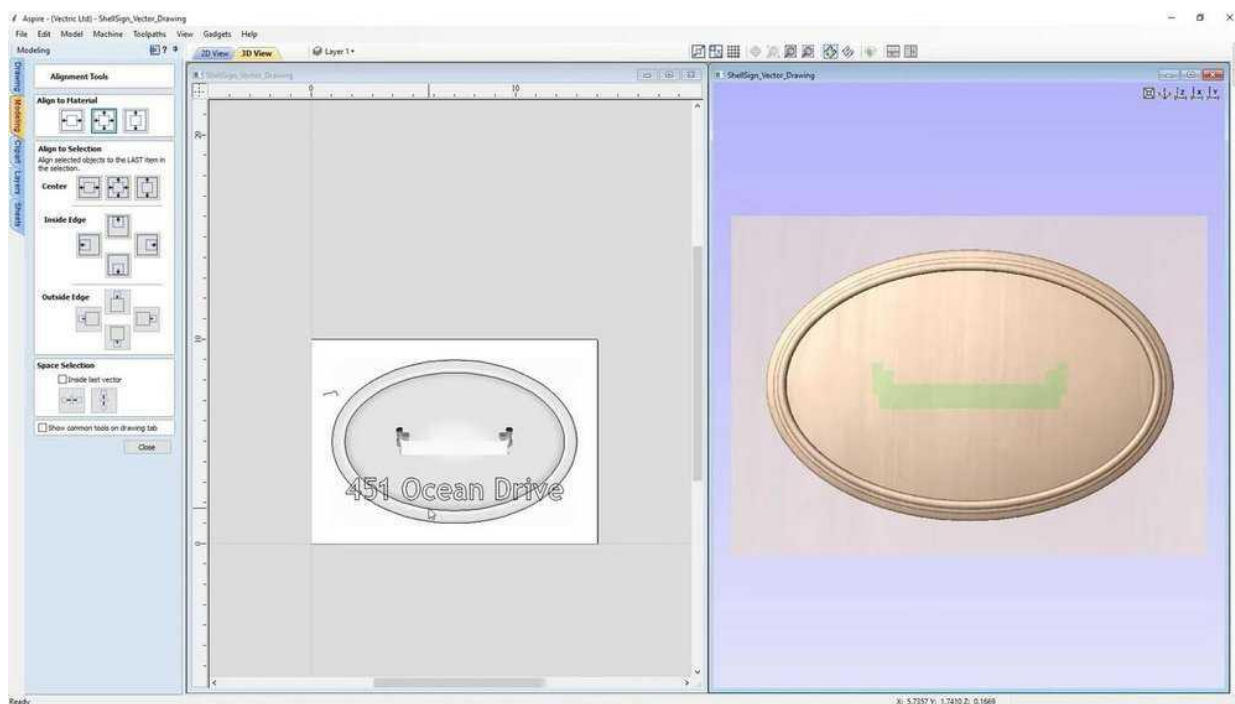


Рисунок 23 – Интерфейс Aspire

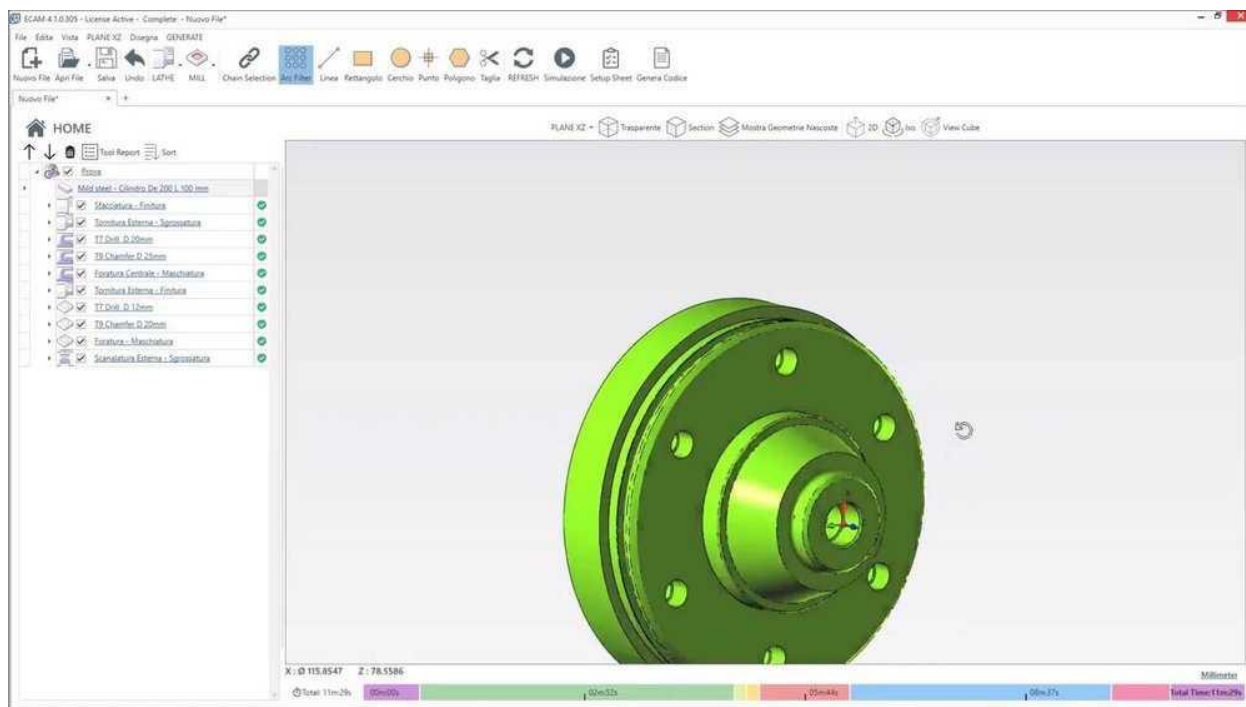


Рисунок 24 – Интерфейс Ecam

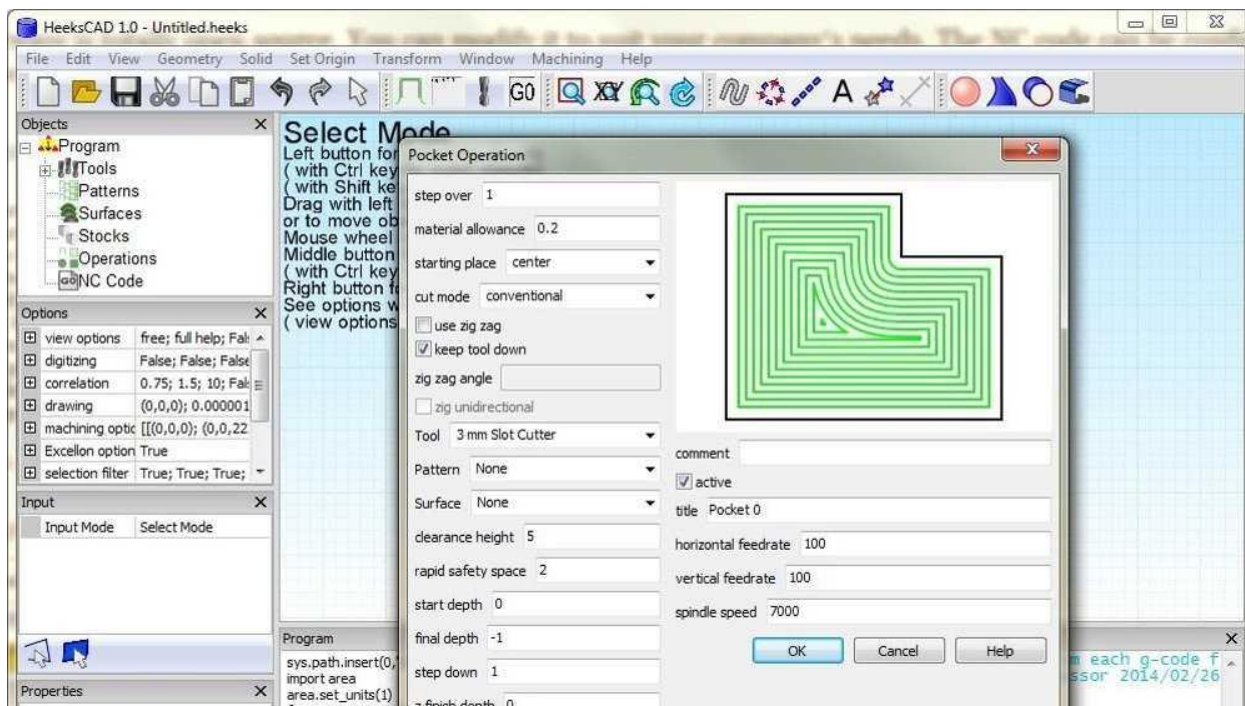


Рисунок 25 – Интерфейс HeeksCNC

Из всех возможных вариантов был выбран бесплатный программный комплекс, имеющий необходимый функционал – Mach3 (Рисунок 26). Это многофункциональный пакет управления станками с ЧПУ. Работает под управлением Windows. Следует сказать, что на сегодня это самый популярный программный продукт для создания систем управления станками с ЧПУ.

## Основные функции, обеспечиваемые ЧПУ:

- Преобразование стандартного ПК в полнофункциональную станцию управления 3-осевым станком с ЧПУ
- Трехмерная графическая визуализация УП G-кодов
- Генерирование файлов УП G-кодов в программе LazyCam или в Мастерах (Wizard)
- Полностью реконфигурируемый перенастраиваемый интерфейс
- Создание пользовательских M-кодов и макросов на основе VB-скриптов
- Управление частотой вращения шпинделя
- Многоуровневое релейное регулирование
- Окно наблюдения за ходом обработки
- Полноэкранный пользовательский интерфейс

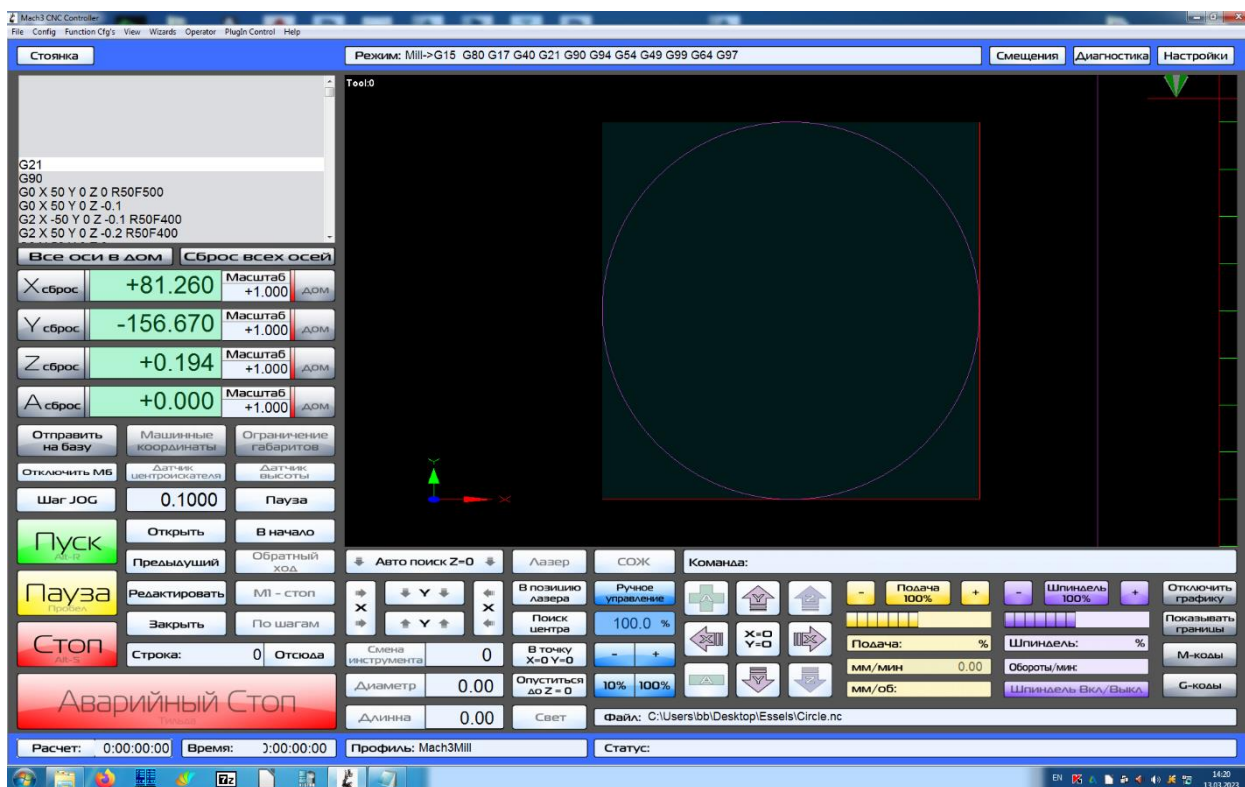


Рисунок 26 – Интерфейс программы Mach3



### 4.3 Пример программного кода

G-code - условное именование языка программирования устройств с числовым программным управлением (ЧПУ). Был создан компанией Electronic Industries Alliance в начале 1960-х. Окончательная доработка была одобрена в феврале 1980 года как стандарт RS274D. Комитет ISO утвердил G-код как стандарт ISO 6983-1:2009.

Производители систем СЧПУ, как правило, используют ПО управления станком, для которого написана программа обработки в качестве готовых команд управления. Краткое описание команд представлено ниже в таблице 11.

Таблица 11 – Описание команд G-code

Коды	Описание
G00-G03	Позиционирование инструмента
G17-G19	Переключение рабочих плоскостей (XY, ZX, YZ)
G20-G21	Не стандартизовано
G40-G44	Компенсация размера различных частей инструмента (длина, диаметр)
G53-G59	Переключение систем координат
G80-G85	Циклы сверления, растачивания, нарезания резьбы
G90-G91	Переключение систем координат (абсолютная, относительная)

Программа, написанная с использованием G-кода, имеет жесткое строение. Все команды управления объединяются в кадры — группы, состоящие из одной или более команд. Кадр завершается символом перевода строки (CR/LF) и имеет номер, за исключением первого кадра программы и комментариев. Первый (а в некоторых случаях еще и последний) кадр содержит только один знак «%». Завершается программа командой M02 или M30. Комментарии к программе размещаются в круглых скобках, как после программных кодов, так и в отдельном кадре.

Порядок команд в кадре строго не оговаривается, но традиционно предполагается, что первыми указываются подготовительные команды (например, выбор рабочей плоскости), затем команды перемещения, затем выбора режимов обработки и технологические команды.

Максимальное число элементарных команд и заданий координат в одном кадре зависит от конкретного интерпретатора языка управления станками, но для большинства популярных интерпретаторов (стоек управления) не превышает 6.

Координаты задаются указанием оси с последующим числовым значением координаты. Целая и дробная части числа координаты разделяются десятичной точкой. Допустимо опускание незначащих нулей, либо их добавление. Также в подавляющем количестве интерпретаторов допустимо не добавлять десятичную точку к целым числам. Например, Y0.5 и Y.5, Y77, Y77. и Y077.0.

Интерпретатор кода (стойка управления) станком запоминает значение введенных параметров и настроек до их смены очередной модальной командой или отмены ранее введенной модальной команды, поэтому необязательно указание в каждом кадре, например, скорости перемещения инструмента.

В приложении Г представлен управляющий код программа для вырезания круга на станке (Рисунок 27).

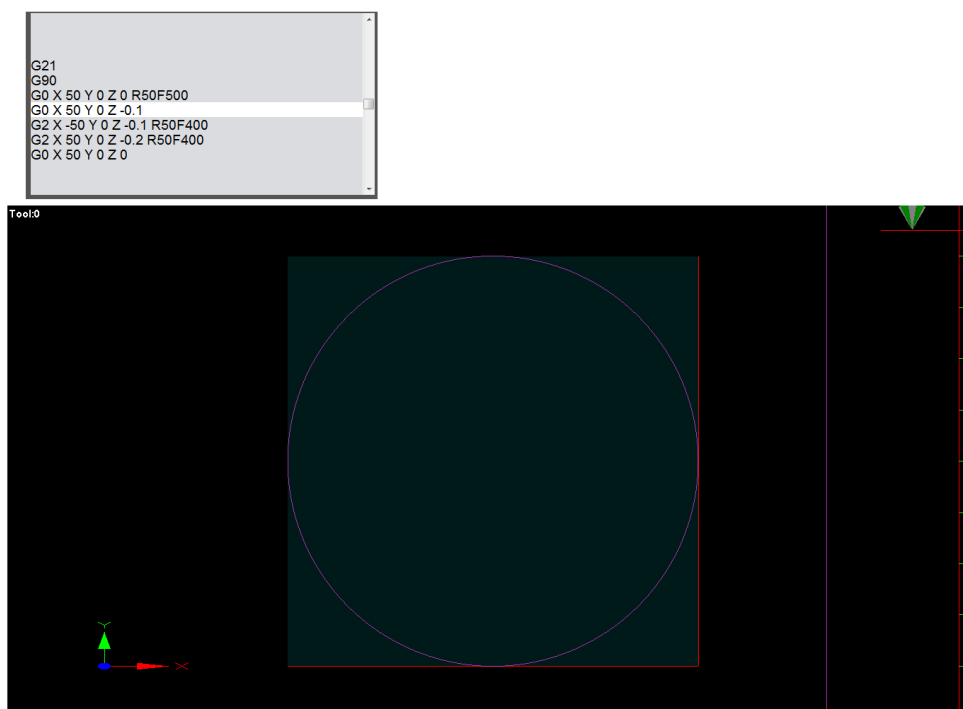


Рисунок 27 – Код программа вырезания круга

## **Заключение**

В ходе написания ВКР, для достижения поставленной цели, были решены все задачи, предусмотренные заданием на ВКР.

Была проанализированы отечественные и зарубежные системы управления СЧПУ серийного производства и программное обеспечение к ним. Для нахождения оптимального варианта были рассмотрены их плюсы и минусы.

Были разработаны структурная информационная модель процесса управления, схемы электрические функциональная и принципиальная прототипа модуля управления.

Был проведен анализ и обеспечен выбор компонентов программного и аппаратного обеспечения.

Результатом ВКР является система управления станком с ЧПУ на микроконтроллерной основе.

Разработанная система управления в дальнейшем будет использоваться для совершенствования учебного процесса.

Верхний уровень системы управления работает на самой распространенной ОС Windows.

По материалам ВКР был сделан доклад на международной студенческой конференции «Перспектив Свободный - 2023» по теме «Разработка и тестирование аппаратно – программного комплекса для фрезерования физических моделей с использованием станка с ЧПУ» рекомендованный к публикации.

## Список используемых источников

1 Стандарт организации. Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. СТУ 7.5 – 07 – 2021 – Красноярск: СФУ, 2021. – 61 с.

2 Федеральный государственный образовательный стандарт 3++ по направлению 09.03.01 "Информатика и вычислительная техника" – 2017. – 13 с.

3 Системы числового программного управления / В. Л. Сосонкин, Мартинов Г.М. – Москва: Логос, 2005. – 296с.

4 Шаговые двигатели и их микропроцессорные системы управления / Т.Кенно Перевод с английского – Москва: Энергоатомиздат 1987. – 200с

5 ДПМ-35 электродвигатели коллекторные для работы в системах автоматизации электронной аппаратуры URL: <http://www.eandc.ru/news/detail.php?ID=22023> (дата обращения: 11.05.2023).

6 Шаговый двигатель дши 200-3 основные технические характеристики URL: <http://www.texnologia.ru/product/hd/dshi-200-3.html> (дата обращения: 11.05.2023).

7 CNC USB controller. Planet CNC URL: <http://www.planet-cnc.com/index.php?page=hardware> (дата обращения: 11.05.2023).

8 AlexGyver библиотека Arduino Uno для ЧПУ станков URL: <https://alexgyver.ru/gyverstepper/> (дата обращения: 11.05.2023).

## ПРИЛОЖЕНИЕ А

### Схема электрическая функциональная разрабатываемой системы управления ЧПУ нижнего уровня

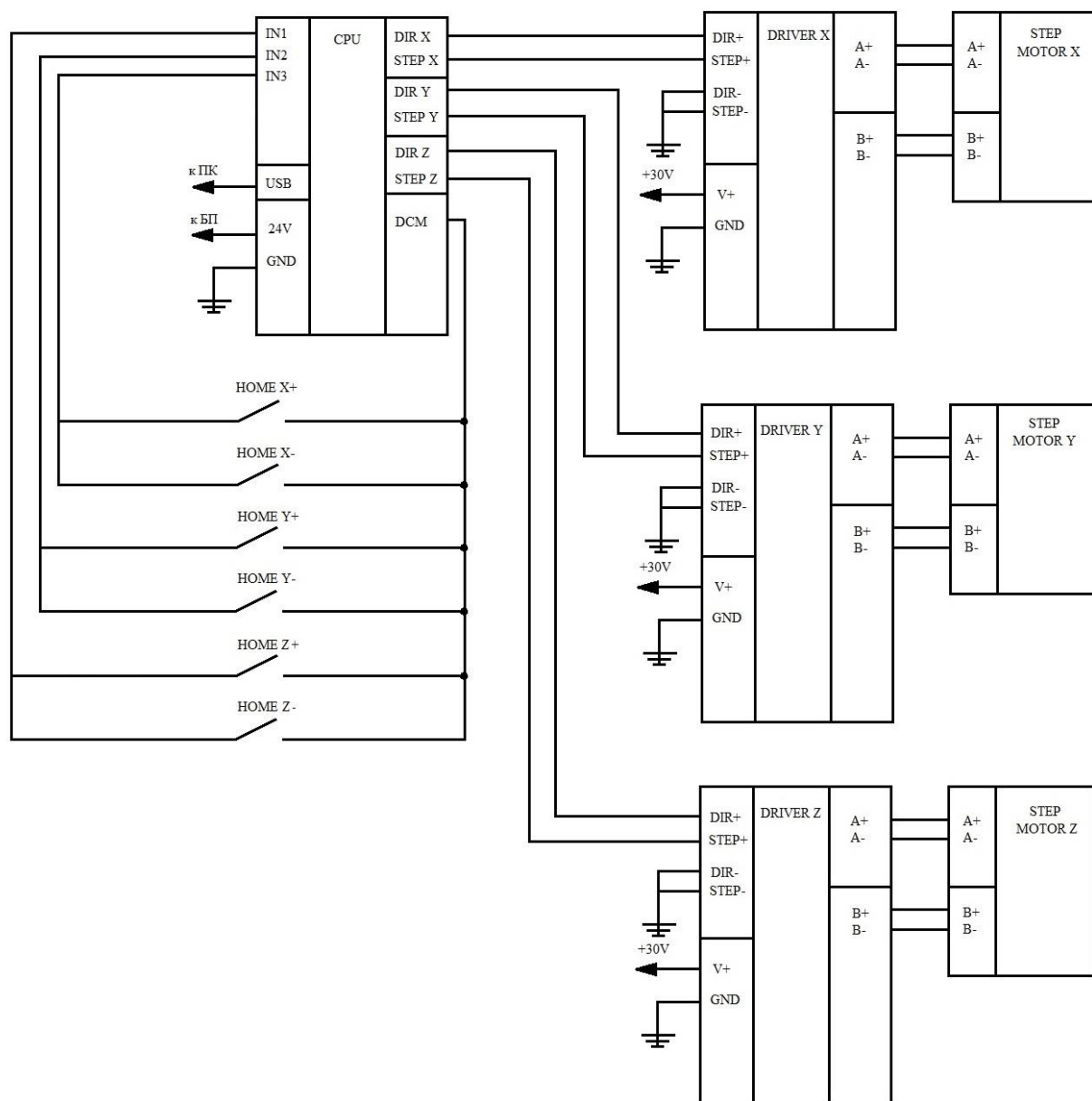


Рисунок А.1 – Схема электрическая функциональная разрабатываемой системы  
управления ЧПУ нижнего уровня

## ПРИЛОЖЕНИЕ Б

### Схема электрическая принципиальная прототипа модуля управления СЧПУ

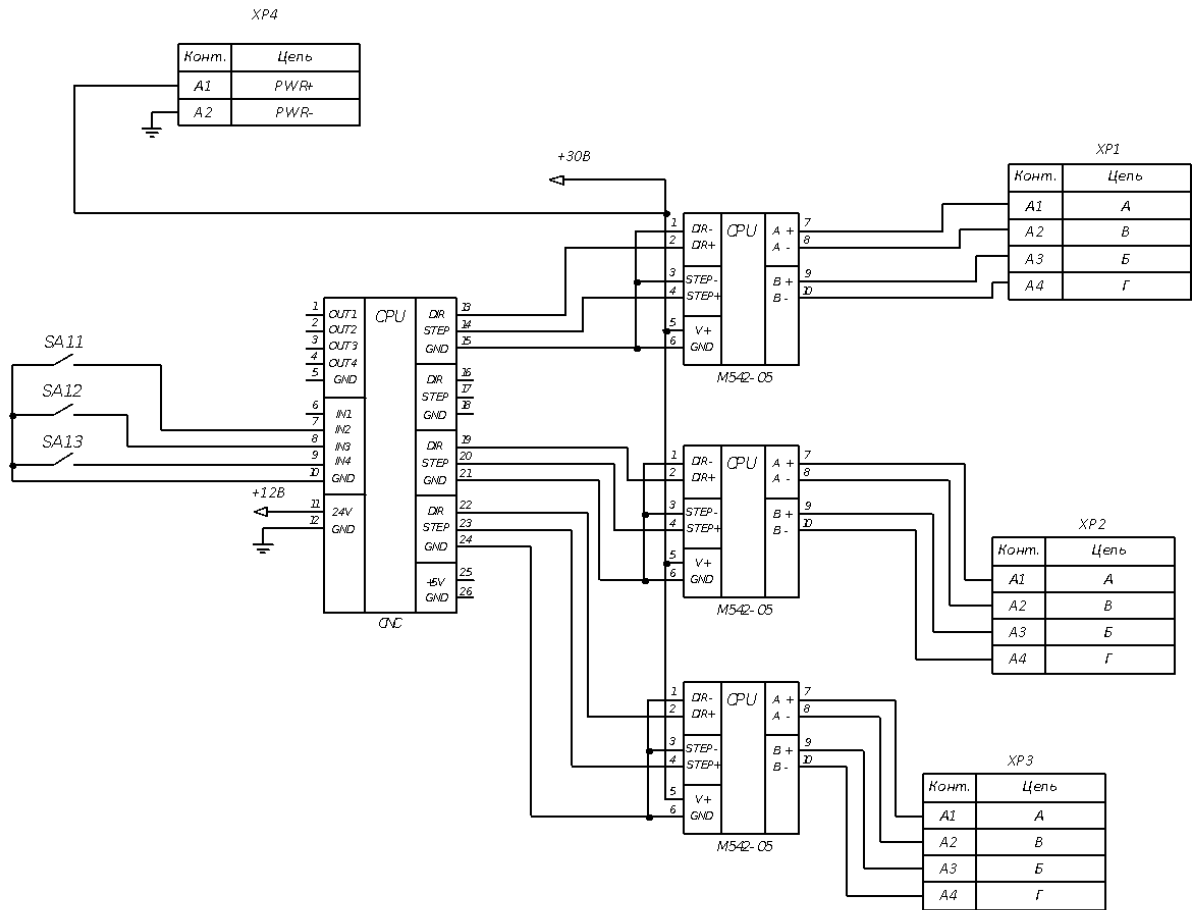


Рисунок Б.1 – Схема электрическая принципиальная прототипа модуля управления СЧПУ

## ПРИЛОЖЕНИЕ В

### Код-программа прототипа системы на Arduino

```
#define GS_NO_ACCEL
#include <GyverStepper2.h>

#define left 42
#define right 44

#define down 46
#define top 48
int i = 0;
int scale = 100;
const int maxSpeed_down = 90;
const int maxSpeed_up = 120;

GStepper2<STEPPER2WIRE> stepperX(25600, 3, 4); //3 - step, 4 - dir
GStepper2<STEPPER2WIRE> stepperY(25600, 6, 7); //6 - step, 7 - dir
//GPlanner2<STEPPER2WIRE, 1> planner; //Провода концевиков:
//черный - Y down; зеленый - Y top
long int target;

void setup() {
  Serial.begin(115200);

  stepperX.setMaxSpeed(maxSpeed_up); // скорость движения к цели
  stepperY.setMaxSpeed(maxSpeed_up);

  pinMode(down, OUTPUT); //Концевик при движении назад
  pinMode(top, OUTPUT); // вперед
```



```

pinMode(left, OUTPUT);
pinMode(right, OUTPUT);

pinMode(8, OUTPUT); //соленоид

digitalWrite(8, HIGH);
// 200к - 16 мм ; 2000к - 156 мм

//goHome(stepperY, top, 1);
//goHome(stepperX, right, 1);
goHome(stepperX, left, -1);
goHome(stepperY, down, -1);

bool dril = false;
int x;
int y;
Serial.setTimeout(-1);
//Serial.write('s');
for(String c = Serial.readStringUntil(';'); c != "0"; c = Serial.readStringUntil(';')){
  //Serial.println(c);
  if(c[0] == 'M'){
    String str = c.substring(2);
    String xstr = "";
    String ystr = "";
    for(int i = 0; i < str.indexOf(' '); i++)
      xstr += str[i];
    for(int i = str.indexOf(' ') + 1; i < str.length(); i++)
      ystr += str[i];
    x = xstr.toInt();

```

```

    y = ystr.toInt();
    if(dril)
        goTo1(x, y);
    else
        goTo0(x, y);
    Serial.write('o');
}
else if(c == "U"){
    dril = false;
    digitalWrite(8, HIGH);
    Serial.write('o');
}
else if(c == "D"){
    dril = true;
    digitalWrite(8, LOW);
    Serial.write('o');
}
else if(c.startsWith("scale")){
    String str = c.substring(6);
    scale = str.toInt();
    Serial.write('o');
}
}
Serial.println("EXIT");
digitalWrite(8, HIGH);
goHome(stepperY, top, 1);
goHome(stepperX, right, 1);
}

void goTo0(long long x, long long y){

```

```

stepperX.setMaxSpeed(maxSpeed_up);
stepperY.setMaxSpeed(maxSpeed_up);
stepperX.setTarget(x*scale, RELATIVE);
stepperY.setTarget(y*scale, RELATIVE);
while(stepperY.tick() + stepperX.tick()){
}
}

```

```

void f(int *x, int *y) {
float orig = *x / float(*y);
int kx = 1;
int ky = 1;
for(int i = 1; i < 99; i ++){
for(int j = i + 1; j < 99; j ++){
if (abs(orig - i / float(j)) < abs(orig - kx / float(ky))) {
kx = i;
ky = j;
}
}
*x = kx;
*y = ky;
}
}

```

```

void goTo2(long long x, long long y){
int kx = abs(x);
int ky = abs(y);
if(abs(x) > abs(y)){
f(&ky,&kx);
stepperX.setMaxSpeed(maxSpeed_down);
stepperY.setMaxSpeed(ky/float(kx) * maxSpeed_down);
}
}

```

```

}
else if(abs(x) < abs(y)){
    f(&kx,&ky);
    stepperY.setMaxSpeed(maxSpeed_down);
    stepperX.setMaxSpeed(kx/float(ky) * maxSpeed_down);
}
else{
    stepperY.setMaxSpeed(maxSpeed_down);
    stepperX.setMaxSpeed(maxSpeed_down);
    kx = 1;
    ky = 1;
}
stepperX.setTarget(x * scale, RELATIVE);
stepperY.setTarget(y * scale, RELATIVE);
while(stepperX.tick() + stepperY.tick()){
}
}

```

```

void goTo1(long long x, long long y){
    double kx = abs(x);
    double ky = abs(y);
    double k;
    if(abs(x) > abs(y)){
        k = ky/kx;
        stepperX.setMaxSpeed(maxSpeed_down);
        stepperY.setMaxSpeed(float(k * maxSpeed_down));
    }
    else if(abs(x) < abs(y)){
        Serial.print("FUCKYOU");
        k = kx/ky;
    }
}

```

```

    stepperY.setMaxSpeed(maxSpeed_down);
    stepperX.setMaxSpeed(float(k * maxSpeed_down));
}
else{
    stepperY.setMaxSpeed(maxSpeed_down);
    stepperX.setMaxSpeed(maxSpeed_down);
}
stepperX.setTarget(x * scale, RELATIVE);
stepperY.setTarget(y * scale, RELATIVE);
while(stepperX.tick() + stepperY.tick()){ }
}

void goHome(GStepper2<STEPPER2WIRE> &stepper, int end, int dir){
    stepper.setSpeed(dir*maxSpeed_up);
    for(bool x = false; !x;){
        if(digitalRead(end) == 1){
            //stepper.setSpeed(0);
            stepper.brake();
            stepper.reset();
            return;
        }
        stepper.tick();
    }
}

void loop() {

}

```

## ПРИЛОЖЕНИЕ Г

### Управляющий код программа для вырезания круга

G21

G90

G0 X 50 Y 0 Z 0 R50F500

G0 X 50 Y 0 Z -0.1

G2 X -50 Y 0 Z -0.1 R50F400

G2 X 50 Y 0 Z -0.2 R50F400

G0 X 50 Y 0 Z 0

