

Министерство науки и высшего образования Российской Федерации Мытищинский филиал
Федерального государственного автономного образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МФ МГТУ им. Н.Э. Баумана)

Факультет Космический

Кафедра «Прикладная математика, информатика и вычислительная техника»

Специальность 09.03.01 «Информатика и вычислительная техника»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА НА ТЕМУ

Разработка ПО для деревообрабатывающего токарно-фрезерного станка с числовым
программным управлением

Выполнил: Аленина М.Ю.

Руководитель: Афанасьев А.В.

Цель

Разработать эффективное и удобное в использовании программное обеспечение для автоматизированной системы управления деревообрабатывающего токарно-фрезерного станка с числовым программным управлением, обеспечивающего управление основными исполнительными механизмами станка, контроль технологических параметров в реальном времени и реализацию практического пользовательского интерфейса.

Актуальность

Современное производство стремится к автоматизации и повышению эффективности, особенно в таких отраслях, как деревообработка. Токарно-фрезерные станки с числовым программным управлением являются ключевым элементом для достижения этих целей, предоставляя возможность создания высокоточных изделий с высокой скоростью и минимальными потерями материала.

Разработанное программное обеспечение может быть непосредственно использовано на деревообрабатывающем станке, что подтверждает практическую ценность данной работы.

Поставленные задачи

- 1) Разработка архитектуры программного обеспечения.
- 2) Реализация функциональных модулей программы в CODESYS.
- 3) Интеграция разработанного ПО с аппаратной частью станка.
- 4) Интеграция и отладка программного обеспечения.
- 5) Разработать алгоритм работы станка.

Инструменты решения поставленных задач

- 1) Язык программирования: Структурированный текст (ST) – текстовый язык, хорошо подходит для написания сложных алгоритмов и структурированного кода. SFC — это графический язык программирования, который особенно хорошо подходит для описания дискретных процессов и процессов, состоящих из последовательных шагов. Его использование позволяет четко и наглядно представить логику работы станка и упростить процесс разработки, отладки и сопровождения программного обеспечения.
- 2) Библиотеки: SysLibFile.lib 20.2.06, OwenLibFileAsync.lib 6.6.14, SYSLIBTIME.LIB 20.2.06, SYSTASKINFO.LIB 20.2.06, SysLibPorts.lib 20.2.06, SYSLIBCALLBACK.LIB 20.2.06, Timer.lib 13.12.08, PruAccessLib.lib 3.10.14
- 3) Отладка и тестирование программы на физическом оборудовании (станке КТФ-8) для выявления и устранения ошибок.

Входные данные

Входные данные поступают в программу в виде текстового файла, который подготавливает оператор станка.

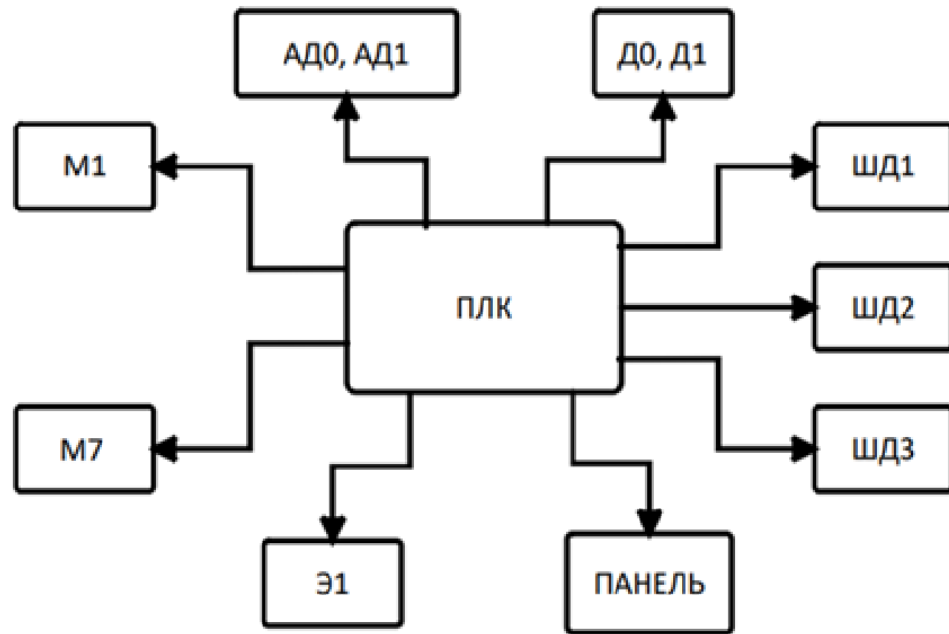
Движение каретки по контуру заготовки описывается следующим образом:

№ символа п/п	Возможное значение	Назначение	Описание
1	0 или 1	Направление движения по оси «Y»	0 – движение «назад» (от центра) 1 – движение «вперед» (к центру)
2,3	01, 02, 03, 04, 05	Перемещение по оси «Y»	в десятых долях мм
4	;	Конец данных для текущего шага	
	*	Конец данных для всей детали	Конец файла

Пример

103; - перемещение вперед на 0,3мм

Архитектурная схема станка КТФ-8



М1, М7 – мотор пилы, мотор фрезера соответственно

АД0, АД1 – концевые выключатели аварийной остановки всех двигателей по оси (X)

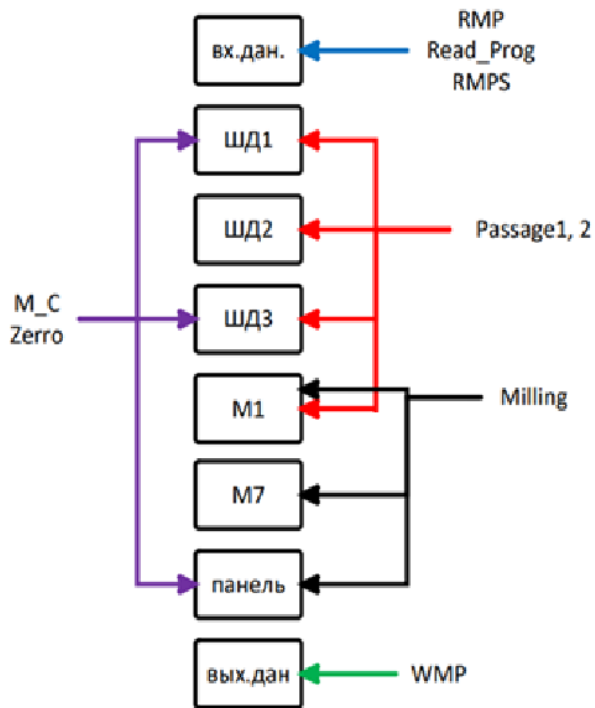
Д0, Д1 – датчики «0» положения продольной оси (X), поперечной оси (Y) соответственно

ШД1, ШД2, ШД3 – шаговые двигатели поперечного движения каретки и заготовки и мотор ходового винта, соответственно

Э1 – энкодер на ходовой винт продольной оси (X)

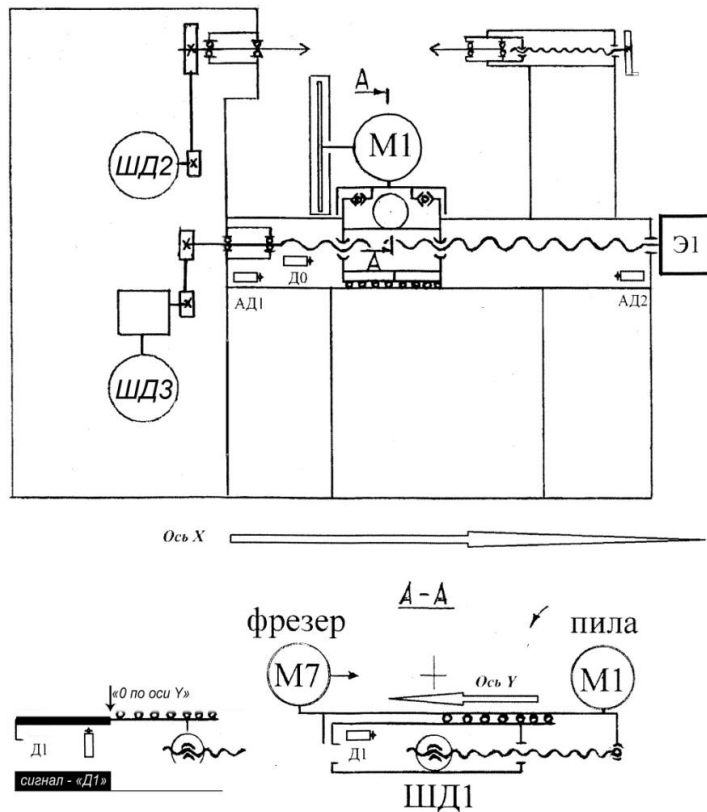
ПАНЕЛЬ – панель оператора

Реализация функциональных модулей программы

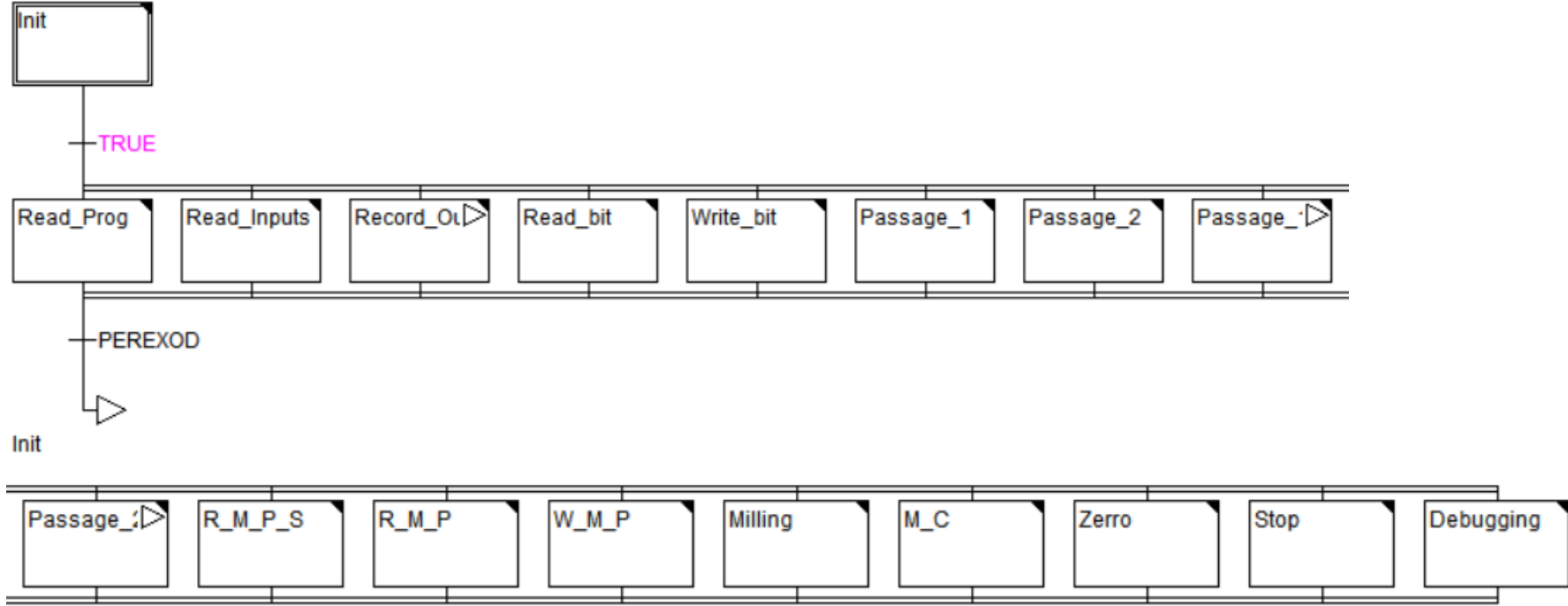


Блоки кода, обрабатывающие данные и отвечающие за работу элементов станка.

Кинематическая схема станка КТФ-8



Функциональная схема ПО



Функциональная схема ПО

Инициализация: начальный этап в SFC. Он содержит процедуры инициализации процесса.

Под начальным шагом мы видим несколько блоков, выстроенных в основной последовательности. Выполнение происходит слева направо. Давайте разберём их:

Read_Prog: Этот блок, считывает данные и значения из памяти программ в ПЛК. Он действует как входной модуль для основного рабочего процесса, в котором данные из других источников будут использоваться в основной программе.

Read_Inputs: этот блок, считывает значения с физических входных модулей ПЛК. Эти значения будут использоваться на последующих этапах обработки.

Record_Outputs: этот блок записывает выходное значение в файл.

Read_bit: это считывание определенного битового значения из входного модуля.

Write_bit: Этот блок устанавливает или сбрасывает конкретное битовое значение в ячейке памяти внутри ПЛК или в выходном модуле.

Passage_1, Passage_2, Passage_3: эти блоки, содержат основной код процесса, поэтому эти блоки представляют собой разделы программы, которые последовательно выполняются как часть основного рабочего процесса.

Использована интегрированная среда программирования Codesys для программирования ПЛК Owen.

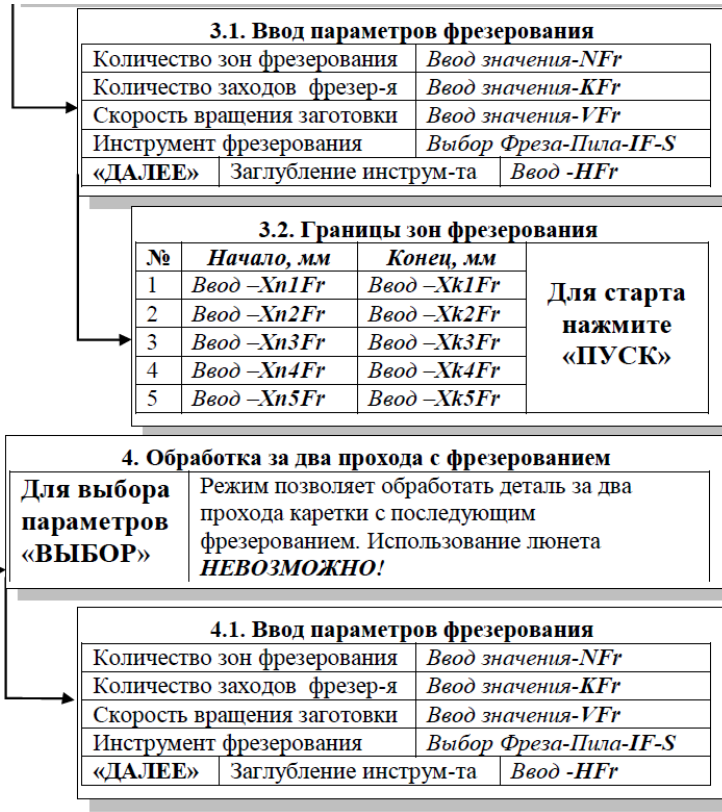
Выбор среды разработки

Интегрированная среда разработки (Integrated Development Environment, IDE) CODESYS, которая является ключевым инструментом для программирования, настройки и отладки систем автоматизации на основе программируемых логических контроллеров (ПЛК).

CODESYS — это универсальная и гибкая интегрированная среда разработки, предназначенная для создания приложений для автоматизации. Она поддерживает множество различных аппаратных платформ, благодаря чему разработчики могут создавать программное обеспечение для широкого спектра устройств: от небольших контроллеров до сложных промышленных систем.

CODESYS предоставляет мощный набор инструментов для проектирования, разработки и отладки систем автоматизации. Рассмотрим основные функциональные возможности : проектирование ПЛК, разработка программного обеспечения, конфигурирование оборудования, отладка и тестирование, управление версиями, разработка и использование библиотек.

Алгоритм работы станка



Алгоритм работы станка

4.2. Границы зон фрезерования			
№	Начало, мм	Конец, мм	Для старта нажмите «ПУСК»
1	Ввод -Xn1Fr	Ввод -Xk1Fr	
2	Ввод -Xn2Fr	Ввод -Xk2Fr	
3	Ввод -Xn3Fr	Ввод -Xk3Fr	
4	Ввод -Xn4Fr	Ввод -Xk4Fr	
5	Ввод -Xn5Fr	Ввод -Xk5Fr	

5. Фрезерование	
Для выбора параметров нажмите «ВЫБОР»	Фрезерование детали, уже прошедшей токарную обработку.

5.1. Ввод параметров фрезерования		
Количество зон фрезерования	Ввод значения-NFr	
Количество заходов фрезер-я	Ввод значения-KFr	
Скорость вращения заготовки	Ввод значения-VFr	
Инструмент фрезерования	Выбор Фреза-Пила-IF-S	
«ДАЛЕЕ»	Заглубление инструм-та	Ввод -HFr

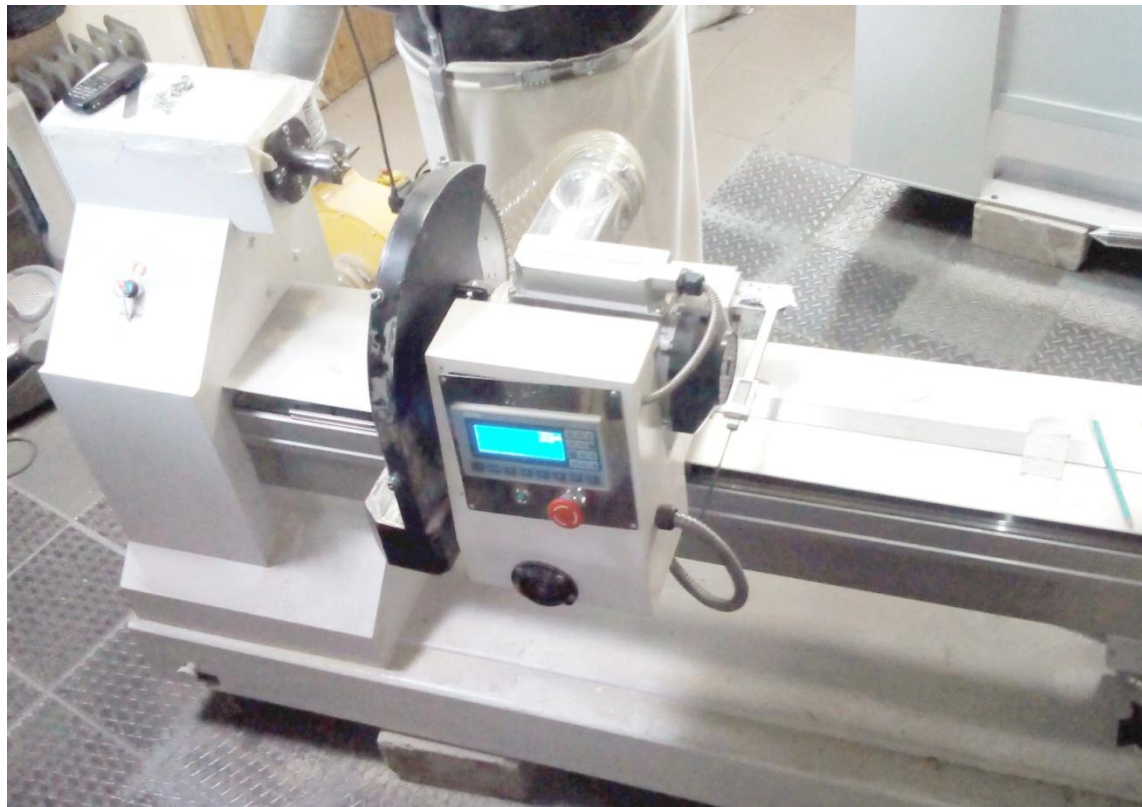
5.2. Границы зон фрезерования			
№	Начало, мм	Конец, мм	Для старта нажмите «ПУСК»
1	Ввод -Xn1Fr	Ввод -Xk1Fr	
2	Ввод -Xn2Fr	Ввод -Xk2Fr	
3	Ввод -Xn3Fr	Ввод -Xk3Fr	
4	Ввод -Xn4Fr	Ввод -Xk4Fr	
5	Ввод -Xn5Fr	Ввод -Xk5Fr	

0. Возврат в положение «0»	
Для старта нажмите «ПУСК»	Возврат каретки в «0» положение. После возврата в «0»-сброс координат X и Y на «0».

8. Загрузка данных контура	
Для старта нажмите «ПУСК»	Происходит загрузка таблицы с координатами профиля заготовки.

9. Ручное управление	
Для старта нажмите «ПУСК»	Ручное управление кареткой осуществляется стрелками. Отображаются координаты X и Y.

Результат работы



Деревообрабатывающий токарно-фрезерный станок с ЧПУ КТФ-8

Результат работы



Продукция выполненная на станке КТФ-8

Заключение

Вывод:

В заключение, разработанное программное обеспечение представляет собой важный шаг в направлении повышения эффективности и расширения возможностей деревообрабатывающего производства. В рамках данной работы были успешно решены ключевые задачи, направленные на создание интуитивно понятного и функционального инструмента для управления токарно-фрезерным станком с ЧПУ. Использование инструментов (Codesys, ПЛК Owen) обеспечило высокую производительность и удобство использования.

Практическая ценность данной работы заключается в предоставлении оператору станка с ЧПУ эффективного инструмента, упрощающего процесс работы и способствующего повышению производительности и качества продукции. Актуальность данной работы подтверждается ее вкладом в автоматизацию и оптимизацию деревообрабатывающего производства, что в конечном итоге способствует повышению конкурентоспособности предприятий в данной отрасли. Разработанное программное обеспечение открывает новые возможности для инноваций в области обработки древесины и является важным шагом на пути к более эффективному и автоматизированному производству.

Заключение

На данный момент в разработке находится:

- 1) Оптимизация части ПО, в которой происходит второй проход обработки заготовки.
- 2) Расчет наилучших теоретических параметров в соответствии с реальными физическими возможностями железа.
- 3) Тестирование и отладка ПО на станке.

Спасибо за внимание