

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт энергетики и электротехники

(институт)

Кафедра Промышленная электроника

11.04.04 – Электроника и нанoeлектроника

(код и наименование направления подготовки, специальности)

Электронные приборы и устройства

(направленность (профиль))

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на тему «Проектирование станка с числовым программным управлением»

Студент

Хаземов А.В.

(И.О. Фамилия)

(личная подпись)

Руководитель

Позднов М.В.

(И.О. Фамилия)

(личная подпись)

Консультанты

(И.О. Фамилия)

(личная подпись)

(И.О. Фамилия)

(личная подпись)

Руководитель программы д.т.н., профессор В.В. Ивашин

(ученая степень, звание, И. О.Ф.)

(личная подпись)

**Допустить к защите**

Заведующий кафедрой

«Промышленная электроника» к.т.н, доцент А.А. Шевцов

(ученая степень, звание, И.О.Ф.)

(личная подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 2016 г.

Тольятти 2016

## АННОТАЦИЯ

Данная магистерская диссертация посвящена разработке и изготовлению станка с числовым программным управлением и состоит из пояснительной записки объемом 94 листах и графической части, объем составляет 8 листов формата А1.

Целью проекта является разработка станка с числовым программным управлением и повышение энергетических характеристик данного проекта станка с минимальным уровнем паразитных резонансных колебаний. А так же низкой себестоимостью при изготовлении.

Задачами проекта являются исследование режимов работы драйвера ШИМ для шаговых двигателей, разработка схемы принципиальной устройства и обоснованный выбор ее элементов, конструкторский расчет.

В первом разделе рассматривается состояние вопроса и приводятся общие сведения об существующих конструкторских решениях станков с ЧПУ, а также анализ выбора конструкторского решения, станка с числовым программным управлением.

Во втором разделе производится разработка станка с числовым программным управлением.

В третьем разделе конструкторско-технологические методы реализации сборочных компонентов устройства.

В четвертом разделе производится технико-экономическое обоснование разработки.

## Содержание

Введение	4
1 Состояние вопроса	
1.1 Обоснование необходимости применения станков с ЧПУ в производстве	7
1.2 Анализ схмотехнических решений станков с ЧПУ	11
1.3 Обоснование и выбор типа мехатронной части ЧПУ станка	13
1.4 Обоснование и выбор типа управления станком с ЧПУ	17
1.5 Анализ и обоснование выбора драйвера управления шаговым двигателем ЧПУ станка	50
1.6 Анализ и обоснование выбора микроконтроллеров для управления ЧПУ станками	61
2 Расчетный раздел	
2.1 Расчет ограничения выходного тока драйвера	80
3 Конструкторско-технологический раздел	81
4 Эффективность проекта	
4.1 Расчет затрат на основные материалы	86
Заключение	90
Литература	91

## Введение

В настоящее время огромное количество промышленных предприятий, в чью сферу деятельности входит механическая обработка, располагают огромным парком станков, имеющие системы числового программного управления (ЧПУ). Эти станки в [1] предназначены для выполнения тех же производственных задач, что и их аналоги с ручным или механическим управлением. Разница между ними лишь в том, что перемещение рабочих органов станков с ЧПУ осуществляется за счет электроники, под управлением специальной компьютерной программы.

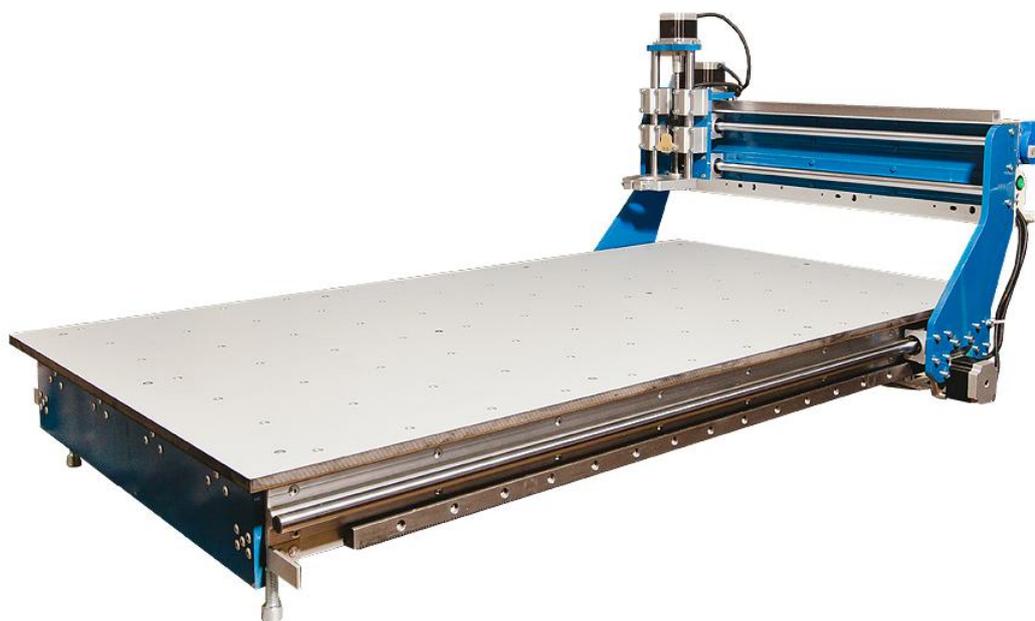


Рисунок 1 - станок с числовым программным управлением

Преимущества использования станков с числовым программным управлением в том, чтобы обеспечении более качественный уровень автоматизации производственного процесса. Изготовление деталей ведется в автоматическом режиме, практически без участия человека, оператора-станочника, роль которого заключается в выполнении операций контроля за процессом

и участия на подготовительном и завершающем этапе: в первоначальной наладке и последующему контролю за ходом выполнения программы и соблюдением автоматикой всех технологических процессов [2]. Автономная работа станков с ЧПУ может продолжаться непрерывно и достаточно долго, причем качество получаемого изделия остается высоким. Таким образом, один человек может одновременно следить и контролировать большой станочный парк с ЧПУ.

Еще одно значительное преимущество – обеспечение производственной универсальности: чтобы станок перешел к изготовлению другого вида продукции, необходимо лишь заменить специально подготовленное программное обеспечение. При этом в любой время можно вернуться к предыдущей программе, уже проверенной в деле. Замену управляющих программ можно осуществлять бесчисленное множество.

Третье немаловажное преимущество – наличие возможности повторять одни и те же действия многократно. Одна и та же программа позволяет производить на требуемом уровне качества тысячи полностью одинаковых по отношению друг к другу деталей. К тому же использование ЧПУ позволяет рабочему механизму выполнять максимально малое количество действий для получения необходимого результата[3]. Как следствие мы получаем максимальную загрузку оборудования и получения максимальной производительности.

Наконец, четвертое преимущество – возможность изготавливать детали достаточно сложной формы, такие, изготовление которых с использованием обычной станочной техники не представляется возможным.

Автоматическое числовое программное управление станком осуществляет промышленный компьютер – важная составляющая станка – при помощи специального программного обеспечения. В программу заложены все параметры обработки и траектории рабочего механизма позволяющие четко соблюдать режимы резания и осуществлять полноценную обработку. Изобретение и последующее совершенствование станков с ЧПУ, без преувеличений, явилось наступлением нового этапа научно-технической революции – ведь ранее управлять станками приходилось исключительно в ручном режимах.

Считывая данные управляющего программного обеспечения, компьютерный модуль посылает на соответствующие двигатели те или иные электронные сигналы. А значит, фактически управляет всеми циклами обработки, поскольку под воздействием двигателей происходит перемещение непосредственно осуществляющих механическую обработку детали рабочими механизмами станка.

# 1 Состояние вопроса

## 1.1 Обоснование необходимости применения станков с ЧПУ в производстве.

Для оценки данного вопроса перечислим все их достоинства и недостатки.

### Технологические достоинства:

- высокая точность обработки;
- сокращение времени производственного цикла;
- сокращение количества операций и переходов;
- сокращение основного и вспомогательного времени;
- устранение трудоемких и специальных приспособлений;
- сокращение доделочных операций;
- улучшение качества, сокращение брака продукции;
- сокращение объема контрольных операций;
- сокращение объема сборки.

### Социальные достоинства:

- физический труд заменяется умственным;
- облегчаются условия труда;
- меняется соотношение между высокооплачиваемым и простым трудом;
- ускоренный процесс получения высокой квалификации.

### Экономические достоинства:

- возможность обработки меньшими партиями;
- ускорение оборачиваемости средств;
- сокращение амортизационных расходов по содержанию основных и вспомогательных помещений;

- сокращение расходов на режущий инструмент и оснастку;
- возможность сокращения общих капиталовложений;
- сокращение прямых расходов;
- повышение коэффициента использования станков;
- фактическая себестоимость равна расчетной.

#### Конструкторские достоинства:

- возможность использовать более сложные поверхности;
- возможность вносить изменения по ходу производства;
- возможность проверки конструкторских решений вычерчиванием по программе.

#### Организационные достоинства:

- улучшение планирования производства;
- сокращение заделов;
- сокращение производственных и вспомогательных площадей;
- сокращение внутрицеховой транспортировки.

#### Потенциальные достоинства:

- возможность сквозной автоматизации;
- использование ЭВМ в производстве;
- создание оптимальных условий труда на всех производственных участках;
- автоматизация инженерно-управленческого труда.

#### Недостатки:

- высокая стоимость станка;
- увеличение сложности ремонта;

- нерациональное использование дорогостоящего станка на простых операциях;
- необходима реорганизация производства;
- требуется более дорогой фундамент;
- требуется переподготовка кадров;
- высокая цена индивидуальной системы управления.

Подводя итог можно сказать что , станки с числовым программным управлением целесообразно применять в следующих случаях:

- в экспериментальном и единичном производстве, при уточнении отдельных элементов конструкции деталей по мере их изготовления, т. е. конструктивные изменения необходимо вносить по ходу изготовления;
- при большом количестве операций, которые выполняются на нескольких станках с большим числом установок заготовки;
- при необходимости изготовления сложных деталей мелкими партиями;
- при изготовлении деталей настолько сложных, что вероятность ошибки работника очень высока (дорогостоящая заготовка, жесткие допуски, высокая точность координат);
- когда стоимость оснастки составляет значительную часть себестоимости детали;
- когда сроки подготовки производства очень малы и специальная оснастка не может быть изготовлена.

## 1.2 Анализ схемотехнических решений станков с ЧПУ:

Общая структура станков с ЧПУ:

- **пульт оператора (или консоль ввода-вывода)**, позволяющий вводить управляющую программу, задавать режимы работы; выполнить операцию вручную. Как правило, внутри шкафа пульта современной компактной ЧПУ размещаются её остальные части;
- **дисплей (или операторская панель)** — для визуального контроля режимов работы и редактируемой управляющей программы/данных; может быть реализован в виде отдельного устройства для дистанционного управления оборудованием;
- **контроллер** — компьютеризированное устройство, решающее задачи формирования траектории движения режущего инструмента, технологических команд управления устройствами автоматики станка, общим управлением, редактирования управляющих программ, диагностики и вспомогательных расчетов (траектории движения режущего инструмента, режимов резания);
- **ПЗУ** — память, предназначенная для долговременного хранения (годы и десятки лет) системных программ и констант; информация из ПЗУ может только считываться;
- **ОЗУ** — память, предназначенная для временного хранения управляющих программ и системных программ, используемых в данный момент.

В роли **контроллера** выступает промышленный контроллер [4], как то: микропроцессор, на котором построена встраиваемая система; программируемый логический контроллер либо более сложное устройство управления — промышленный компьютер.

Важной характеристикой CNC-контроллера является количество осей (каналов), которые он способен синхронизировать (управлять) — для этого требуется высокая производительность и соответствующее ПО.

В качестве исполнительных механизмов используются сервоприводы, шаговые двигатели.

Для передачи данных между исполнительным механизмом и системой управления станком обычно используется промышленная сеть (например, CAN, Profibus, Industrial Ethernet). В более простых станках шкаф управления монтируется в [5] непосредственной близости от исполнительных устройств.

Управляющая система считывает инструкции специализированного языка программирования программы, который затем интерпретатором системы ЧПУ переводится из входного языка в команды управления главным приводом, приводами подач, контроллерами управления узлов станка (например, включить/выключить подачу охлаждающей эмульсии).

Разработка управляющих программ в настоящее время выполняется с использованием специальных модулей для систем автоматизированного проектирования(САПР) или отдельных систем автоматизированного программирования (CAM), которые по электронной модели генерируют программу обработки.

Наиболее распространенный язык программирования ЧПУ для металлорежущего оборудования описан документом ISO 6983 Международного комитета по стандартам и называется «G-код». В отдельных случаях — например, системы управления гравировальными станками — язык управления принципиально отличается от стандарта. Для простых задач, например, раскрыя плоских заготовок, система ЧПУ в качестве входной информации может использовать текстовый файл в формате обмена данными — например, DXF или HPGL.

**DXF** (англ. *Drawing eXchange Format*) — открытый формат файлов для обмена графической информацией между приложениями САПР. Был создан фирмой Autodesk для системы AutoCAD. Поддерживается практически всеми CAD-системами на платформе PC.

**HPGL** (иногда *HP-GL*) является основным языком управления принтерами, используемым плоттерами Hewlett-Packard. Его название представляет собой аббревиатуру *Hewlett-Packard Graphics Language*. В данный момент он является стандартным почти для всех плоттеров.

### 1.3 Обоснование и выбор типа мехатронной части ЧПУ станка

#### 1.3.1 Портальный ЧПУ станок

Портальная конструкция ЧПУ станка показана на рисунке 1.3.1. она хороша простотой исполнения и свой первый самодельный ЧПУ станок обычно начинают с нее. Фрезерная часть на ЧПУ станках портального типа может двигаться в осях Z и X.

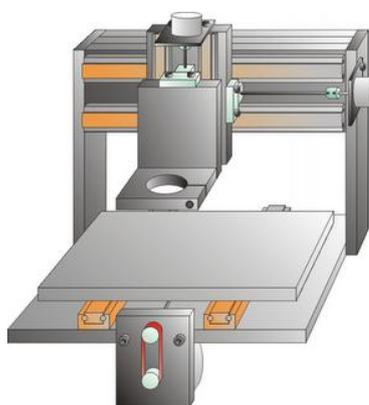


Рисунок 1.3.1 - Портальный ЧПУ станок

Достоинства конструкции портального типа:

- Высокая жесткость на изгибы при обработке.
- Простота изготовления.

Недостатки ЧПУ станка портального типа:

- Нет возможности обработки тяжелых заготовок, так как заготовка держится и перемещается по оси X, нагружая ее своим весом.
- Размер обрабатываемой детали ограничен размером портала

### 1.3.2 Станок ЧПУ с движущимся порталом

Станок второго типа показан на рисунке 1.3.2, он сложнее станка портального типа. Его часто рекомендуют для самостоятельного изготовления ЧПУ станков. Фрезерная часть ЧПУ станка такого типа может двигаться в 3-х осях.

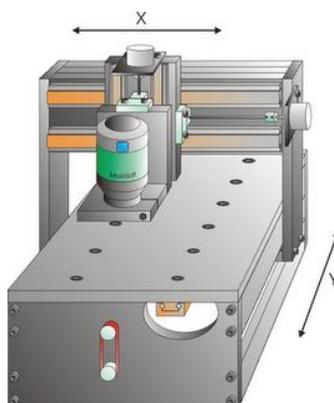


Рисунок 1.3.2 - Станок ЧПУ с движущимся порталом

Достоинства:

- Простота изготовления.
- Вес обрабатываемой детали ничем не ограничен.

- Удобен электронных печатных плат, особенно поточным методом.
- Можно обрабатывать заготовку неограниченной длины по оси Y.

Недостатки:

- Нужно использовать жесткую и прочную направляющую для оси X, так как именно эта будет испытывать большие нагрузки при обработке.

### 1.3.3 Движущаяся фрезерная часть по оси Z

Третий тип конструкции показан на рисунке 1.3.3. Обычно используется при доработке промышленных образцов (например ручного фрезерного станка Proxon) или переделки из сверлильного станка в гравировально-фрезерный ЧПУ станок.

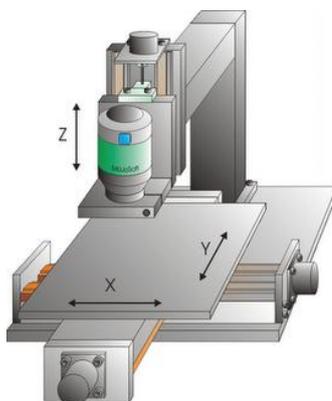


Рисунок 1.3.3 - Станок ЧПУ с движущейся по оси Z фрезерной частью

Достоинства:

- Простота переделки из готовой конструкции. Координатный стол для ЧПУ станка такого типа можно взять готовый и вся работа будет заключаться только в установке двигателей и электроники.

Недостатки:

- Нет возможности обработки тяжелых заготовок.
- Ограниченный размер обрабатываемой заготовки. Чаще всего координатные столы такой конструкции применяют для фрезеровки небольших деталей и размер рабочего поля ЧПУ станка составляет 15x15 см
- При обработке краев заготовки возможен прогиб осей X, Y - так как вес заготовки смещенный в один край будет изгибать оси.

### 1.3.4 Безпортальный ЧПУ станок

Четвертый тип станка представленный на рисунке 1.3.4, гораздо сложнее в изготовлении предыдущих и не рекомендуется для первого опыта в изготовлении самодельного ЧПУ станка. Фрезерная часть такого станка может двигаться во всех направлениях.

Сложность данной конструкции в жесткости связки осей X и Y. Добиться того, чтоб при обработке на максимальном удалении по оси Y не было смещений или искривления оси вызванных сопротивлением материала обработке - очень сложно. Особенно если строить из подручных материалов. Из-за этого может пострадать точность обработки.

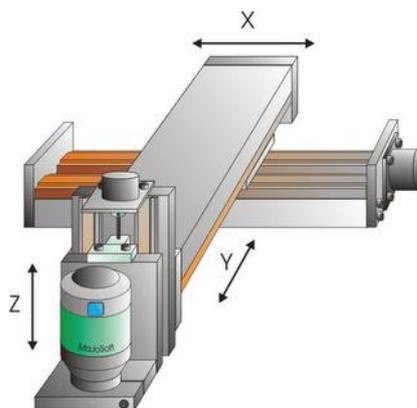


Рисунок 1.3.4 - Безпортальный ЧПУ станок

Достоинства:

- Можно обрабатывать заготовки любых размеров. Достаточно просто

переставлять CNC станок на заготовке. Даже если оси сделаны всего 20 см можно обрабатывать заготовку 5х5 метров или больше!

- Вес обрабатываемой детали ничем не ограничен.

Недостатки:

- Сложно сделать жесткую конструкцию. Чем длиннее ось Y тем больше рычаг и сильнее искривление оси при обработке заготовки.

Рассмотрев все типы исполнения мехатронной части станков, мы выбрали станок ЧПУ с движущимся порталом. Он наиболее полно отвечает требованиям у изготавливаемому станку.

#### 1.4 Обоснование и выбор типа управления станком с ЧПУ

Исходя из информационного признака и по количеству потоков информации для управления станком с числовым программным управлением используется замкнутый и разомкнутый тип. Рисунок 1.4.1.

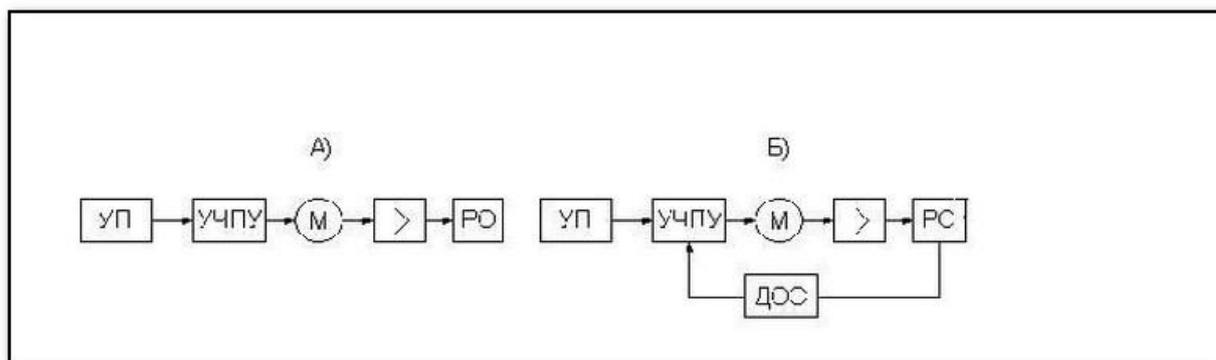


Рисунок 1.4.1 - Типы управления станком ЧПУ а) разомкнутое , б) замкнутое

В случае разомкнутого управления [б] существует только один информационный поток, который направляется от УП (управляющий поток)

на РО (рабочий орган). Станки, имеющие такое управление, оснащены шаговыми двигателями ШД, мотором М.

В случае замкнутого управления существует два информационных потока: один идет от УП, другой — от ДОС (датчиков обратной связи).

Конструктивно наиболее сложными являются замкнутые системы управления, однако они и функционируют точнее последних, поскольку в них фактическая отработка передвижений сравнивается с заданной, а также выполняется коррекция УЧПУ по информации датчиков обратной связи

В системах управления замкнутого типа выделяются адаптивные или самонастраивающиеся системы. Такие системы содержат дополнительные информационные потоки, способные корректировать процедуру обработки в зависимости от реальных условий [7] (колебания припуска, затупление инструмента и пр.).

Рассмотрев типы управления станков с ЧПУ. И проанализировав задачи, которые будет выполнять хоббийный ЧПУ станок. Мы выбрали разомнутый тип управления, отвечающий всем нашим требованиям. Применив в позиционировании шпинделя шаговые двигатели. Благодаря практически полной безинерционности шаговых двигателей и огромного крутящего момента, возможно полностью отказаться от датчиков обратной связи, что существенно упростит конструкцию станка.

Разрабатываемый автономный чпу станок будет работать под управлением специального программного обеспечения, которое должно уметь считывать параметры и траектории обработки разработанной заготовки. Для этого необходимо выбрать на каком языке будут писаться все необходимые параметры.

На сегодняшний день существует более 100 языков для написания программ станкам с ЧПУ, но до сих пор не существует единого языка, который в достаточной мере удовлетворял бы всем необходимым требованиям. Языки отличаются степенью автоматизации и степенью специализации, самый распространенный язык G-код.

G-код — условное наименование языка для программирования устройств с числовым программным управлением. Этот язык был создан компанией Electronic I.A. в конце 1950-х , начале 1960-х. Окончательная доработка и одобрение произошло в феврале 1980 года и было принято стандартом RS274D. Комитет стандартов ISO принял G-код стандартом ISO 6983-1:2009. Государственный комитет по стандартам СССР принял его как ГОСТ 20999-83. В советской нормативной документации G-код именуется кодом ИСО 7-бит ( ISO 7 bit ). Создан для передачи информации ЧПУ в виде кода написанного машинным языком, аналогично как и коды PC8C или AEG.

Изготовители систем с числовым программным управлением, применяют программное обеспечение управления станком, для которого написана программа обработки деталей в качестве основной команды управления [8], G-код используется в виде основного языка программирования, увеличивая его на своё усмотрение. Ниже представлено подробное описание G-кода с примерами.

#### Описание G-кода

G00 –перемещение инструмента при максимальной скорости. Код G00 применяется для перемещения рабочего механизма от одного обрабатываемого участка к другому. Применяется для максимально быстрого перемещения инструмента для резки к искомому месту обработки детали или к позиции безопасности. Для выполнения обработки детали никогда не применяется ускоренное перемещение инструмента, потому что скорость движения исполнительного механизма станка очень довольно быстрая и

меняется на протяжении всей работы . Код G00 сменяется другой командой при применении следующих кодов: G01, G02, G03.

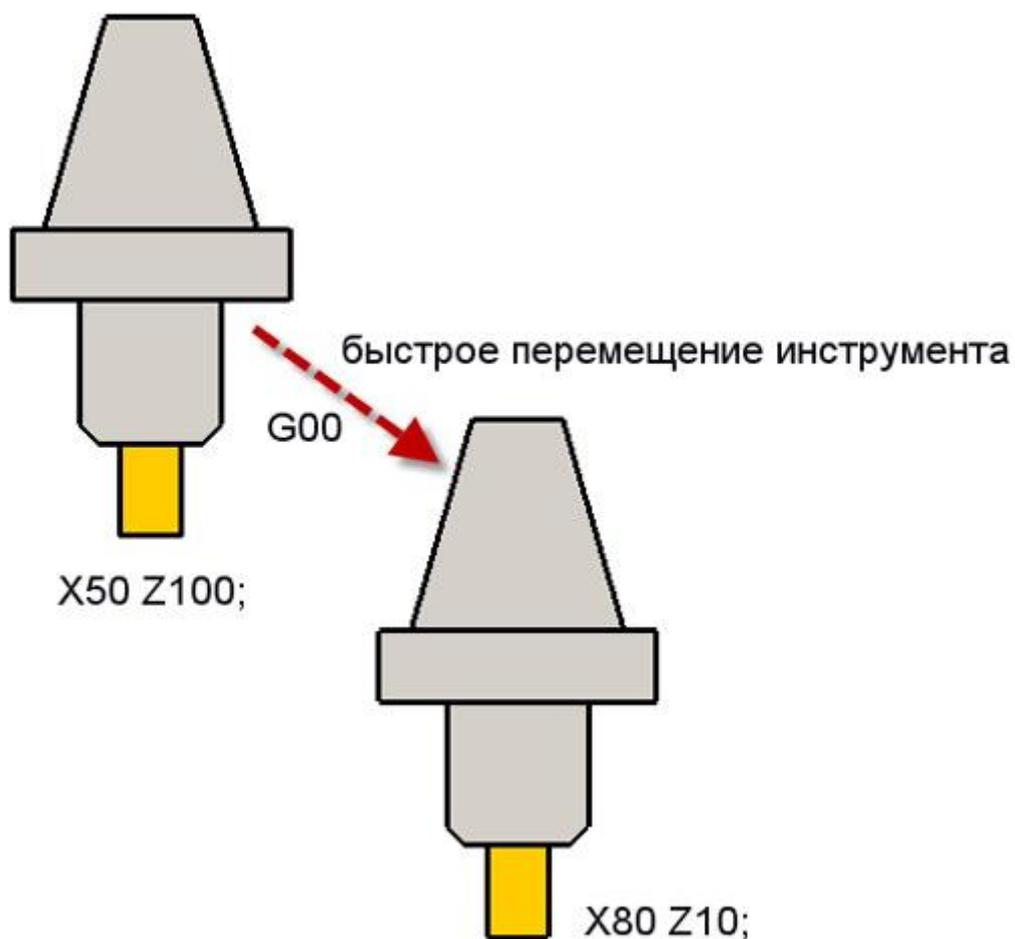


Рисунок 1.4.2 - Пример ускоренного перемещения инструмента. G00 X80 Z10 – перемещение в точку с координатами ( 10; 80 ).

G01 – линейная интерполяция. Код G01 – исполнительная команда позволяющая осуществить перемещение исполнительного механизма по прямой с заранее установленной скоростью. Скорость перемещения инструмента задается адресом - F. Код G01 отменяется с помощью кодов G00, G02HG03.

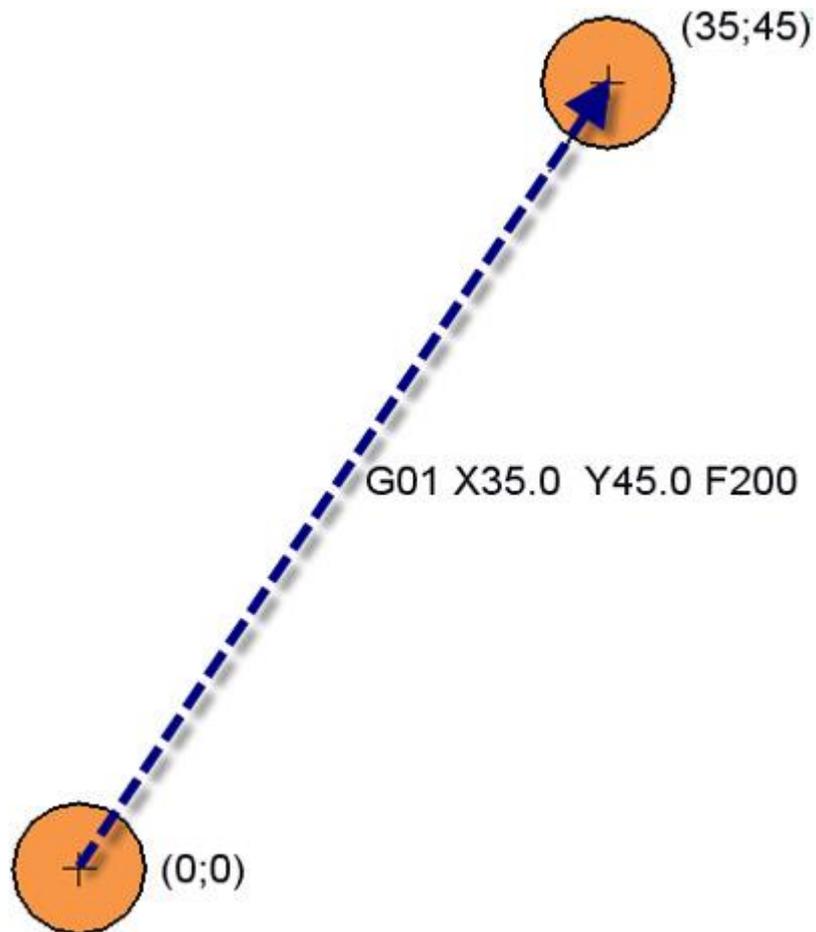


Рисунок 1.4.3 - Пример обеспечения линейной интерполяции инструмента. G01 X35 Y45 F200 – перемещение инструмента в точку с координатами (35; 45) по прямой со заданной скоростью подачи 200 миллиметров в минуту.

G02 – круговая интерполяция (дуга в направлении по час. стр.). Код G02 используется для исполнения интерполяции с кругом, иными словами для перемещения инструмента по кругу в направлении часовой стрелки с заранее заданной скоростью хода. Скорость перемещения заранее принимается адресом - F. Код G02 отменяется кодами G03, G00 , G01.

G03 – перемещение по кругу (против час стр.). Код G03 используется для перемещения исполнительного инструмента по кругу, иными словами для позиционирования исполнительного инструмента по кругу против часовой стрелки с заранее установленной скоростью. Скорость перемещения задается адресом - F. Код G03 снимается кодами G02, G01 ,G01.

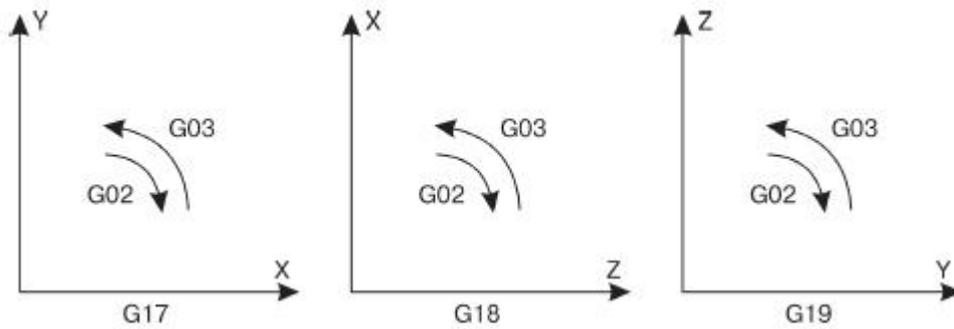


Рисунок 1.4.4 - Интерполяция в круге на различных плоскостях

G04 – пауза. Исполняемый параметр задает выдержку с интервалом времени задаваемым заранее выполняется кодом G 04 . Этот программный код выполняется вместе с P - а так же адресом - X , он задает времени паузы. Время находится в интервале между 0.001 и 99999.999 секундами. Код G04, P - или адрес - X выполняются в одном заходе , он не должен иметь каких либо перемещений по оси координат.

Когда необходимо определить времени задержки, используется команда P, в этом случае нельзя программно использовать десятичную точку. Адрес с индексом P назначает выдержку по времени в миллисекундах, а индекс X назначает выдержку в секундах. Когда командный код G04 устанавливается без времени задержки импульса, она принимается станком как команда для точной остановки инструмента.

Пример:

G04 X1.6 – задержка 1.6 секунды;  
G04 P2500 – задержка 2.5 секунды.

G09 – остановка точная. Из-за увеличения скорости и соответственно её замедления относительно осевых переходов исполнительных механизмов станка , отсутствует точная выполнение среза угловых кромок при перемещении команды резания от одной к другой. Этот недочет обработки изделия выглядит как в притуплении и закруглении углов.

Допустим, когда вы обрабатываете по периметру контур в виде прямоугольника и пытаетесь сделать кромку в углу очень острую (рис. 6). Работая в обычном режиме, наверняка, что при перемещении от движения по оси X к перемещению по оси Y выйдет скругление кромок (рис. 7). Очень сильно этот вредный эффект виден когда станок работает при очень высоких скоростях подачи и при работе в огромных центрах обработки.

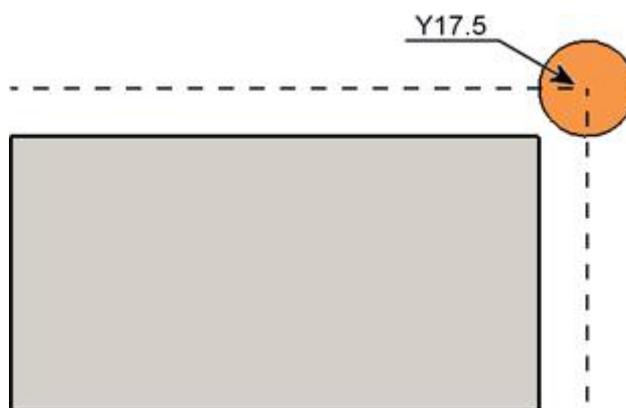


Рисунок 1.4.5 - Получение острой кромки в правом углу

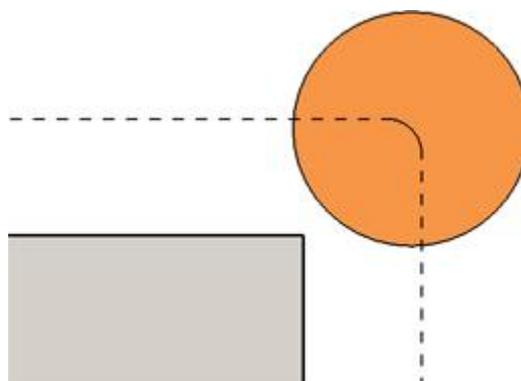


Рисунок 1.4.6 - Обрезка методом скругления кромки

G09 это код немодальный, он используется когда необходимо согласование текущей траектории инструмента станка с заданной заранее траекторией. При перемещении из одного движения к другому станок совершит конечное перемещение в заданную позицию.

G09 указывается вместе с заданной координатой, для которой будет выполнено точное позиционирование. Управляющая программа,

гарантирующая получение острой кромки угла прямоугольного контура, будет выглядеть так:

```
%  
  
O0005  
  
N100 G21  
  
N102 G0 G17 G40 G49 G80 G90  
  
N104 T1 M6  
  
N106 G0 G90 G54 X30. Y-22.6 S1000 M3  
  
N108 G43 H1 Z100.  
  
N110 Z10.  
  
N112 G1 Z-2. F90.  
  
N114 Y-12  
  
N116 G09 Y17.7  
  
N118 X-25.  
  
N120 X-35.  
  
N122 Z8.  
  
N124 G0 Z90.  
  
N126 M5  
  
N132 M30  
  
%
```

Когда инструмент приходит в координату Y17.7, то станок выполняет точную остановку. Время задержки в этой позиции определяется значением параметра системы.

G10 – ввод данных управления станок. Команда G10 устанавливает и изменяет систему координат в работе и вводит заданные параметры для корректировки инструмента станка с помощью программы управления.

При желании внести иные параметры для корректировки с помощью программы управления, их необходимо разместить в начале кода. Таким образом происходит согласование параметров корректировки и изменения в программе управления.

Как правило, для ввода параметров корректировки используется следующий вид кода:

```
G10 L11 P_ R _ ;
```

где G10 – режим ввода параметров ; L11 – настройка корректировки инструмента станка ; P – выбор корректировки , который надо изменить; R – вносимое значение.

Если команда G10 выполняется вместе с командой G90, то параметры в регистрах корректировки будут изменены. Если G10 работает совместно с командой G91, то параметры в корректорах складываются или вычитаются с значением R. Например , G10 G90 L11 P13 R91.14 меняет действительное значение в корректировке № 13 на новое значение 91.14.

При необходимости установить или сместить рабочую координатную систему, выполняется следующая команда:

```
G10 L2 P _ X _ Y _ Z _ ;
```

где G10 – режим ввода данных , его включение; L2 – установка рабочей стандартной системы координат; P – инициализация системы координат; X, Y, Z – параметры, выбирающие заданное расположение рабочей системы координат.

Предварительное значение G10 будет модальной и будет ею до того времени, пока не поступит команда отмены G11. Прежде чем использовать G10 необходимо внимательно изучить документацию к устройству, потому что параметр G10 бывает разнообразным по значению.

G11 – означает, окончание режима ввода данных в станок. С помощью данной команды G11 прекращает свое действие G10 для начала режима ввода параметров в станок.

G15 – отменяет режим ввода данных полярных координат. С помощью команды G15 вы деактивируете установку в полярной системе координат и переходите назад к вводу данных с помощью систему координат в прямоугольной форме.

G16 – включает режим полярных координат. Данная команда G16 будет исполняться в полярной системе координат. В связи с чем применяется позиционирование определяемое зависимостью расстояния и угла от точки нуля данной координатной системы или от текущей позиции инструмента.

Использование станка в полярной системе координат разрешается в каждой из трех осей системы. Кодом программы G17 вы осуществляете перемещение в осях XY, с кодом программы G18 – в осях XZ, а при помощи кода программы G19 – в осях YZ.

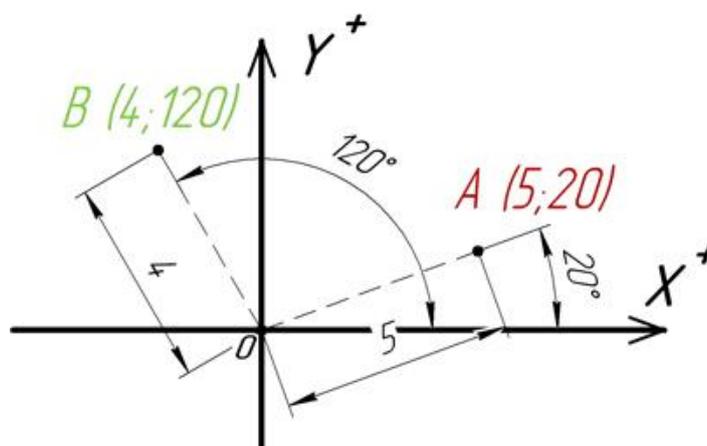


Рисунок 1.4.7 - Полярные координаты: точка A (5;20) и точка B (4; 120)

Когда выбирается данная плоскость осей XY, то X - адрес указывает на радиус, а Y выбирает относительно оси X его угол. Когда используется ось XZ, то адрес X обозначает радиус, а адрес Z обозначает относительный угол оси X. Когда заданная ось YZ, то адресное поле - Y задает радиус, а Z задает относительный угол оси Y. Угол отсчитываемый против часовой стрелки, является положительным.

Перемещения в полярной системе координат, на которые указывают действующая команда G90, исполняются строго относительно нулевой точки настоящей системы координат в работе. При исполнении кода программы G91, перемещения в полярной системе координат исполняются с отношением к текущему местоположению. Как абсолютные или относительные переменные возможно задавание параметров углов и радиусов. Полярное перемещение определяется углом нулевой точки системы в работе и расстоянием от реального расположения инструмента станка.

Исключить преобразование полярных координат в прямоугольные позволяет подготовительная функция

G16

G90 G17 G16

G81 G98 X4 Y30 Z-2 R0.7 F45

Y60

Y90

G15 G80

Код программы G16 есть модальная функция, исходя из этого она действует до того времени, пока не выключится кодом G15.

G17 – выбор оси XY. Предварительный код G17 необходим для оси XY в качестве настоящей действующей оси. Ось XY становится рабочей при для

исполнения интерполяции в круговом формате, изменения координатной системы и циклов обработки детали в круговой проекции.

G18 – команда выбора плоскости оси XZ. Предварительный код G18 нужен для оси XZ устанавливаемым в параметре текущей оси. Ось XZ устанавливается главной при применении круговой интерполяции [9], круговом вращении координатной системы и непрерывных циклов обработки деталей.

G19 – команда выбора оси YZ. Предварительная команда G19 необходима для оси YZ устанавливаемой в качестве текущей. Ось YZ устанавливается основной при применении круговой интерполяции и путем постоянных циклов обработки с вращением системы координат.

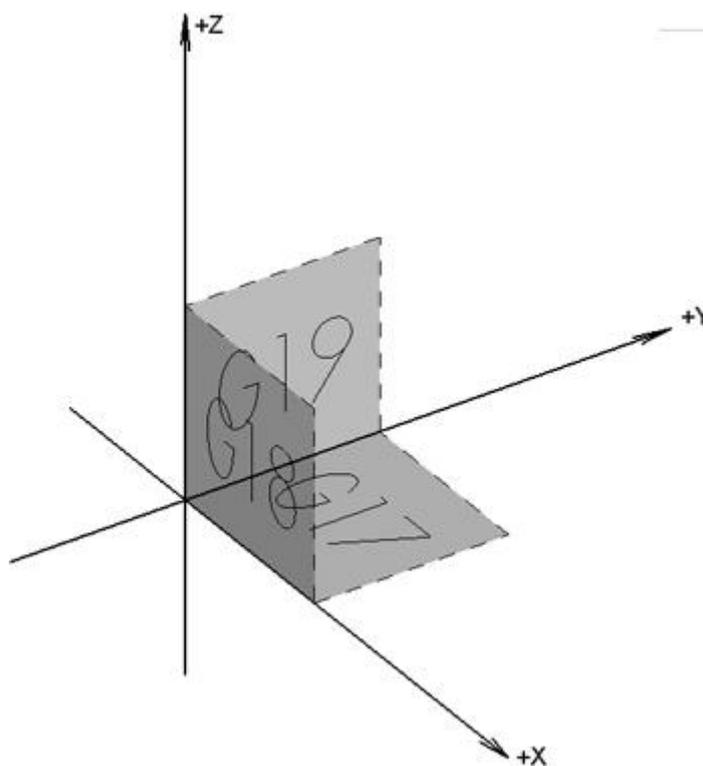


Рисунок 1.4.8 - G19, G18, G17 применяются для активации системы координат в плоскости

G20 – код для ввода параметров в дюймах . Команда G20 использует работу станка с дюймовыми значениями перемещений. Если используется данный параметр, все вносимые параметры устанавливаются как дюймовые

значения. Рекомендации во всех циклах, используемых в дюймовых системах, поставить исполняемую команду G20 в начало кода программы, чтобы в случае, если в программе, исполненной до этого, было включено метрическое вычисление, необходимо задать выбор правильный формат выбора.

Пример:

N10 G20 G40 G49 G54 G80 G90 – код G20 в безопасной строке. Код является модальной функцией и исполняется до поступления кода G21.

G21 – введение параметров в метрической системе координат. Код G21 включает режим работы в метрических параметрах системы. Если включен данный режим, все исполняемые переменные распознаются метрическими. В исполняемых программных кодах, написанных в метрической системе, желательно использовать команду G21 в начале программы, чтобы в случае, когда в программном коде, исполненном до этого, применялось дюймовое вычисление, использовать правильный формат выбора.

Пример:

N10 G21 G40 G49 G54 G80 G90 – команда G21 в безопасной строке. Данный код есть модальная функция и исполняется до команды G20.

G22 – использование условия перемещений до предела. Команда G22 включает предел значений для установки. В данном случае инструмент станка не уходит за область ограниченную пределами[10]. Данные пределы устанавливаются значениями введенными в ЧПУ.

G23 – дезактивация режима предельного позиционирования станка при перемещении. Используя команду G23 заданные параметры перемещения не учитываются. Команда G23 выключает действие команды G22 и осуществляет перемещение инструмента в различные рабочие зоны координатной сетки.

G27 – команда проверки возвращения к стартовым позициям. Команда G27 исполняется также как команда G28. Различие лишь в том, что позиция, в которой переместился рабочий инструмент, не является заданной ранее позицией, тогда случае команды G27 управляющая электроника станка сигнализирует об аварии и выдает аварийный сигнал.

Команды G27 и G28 используются в циклических программных макросах смены инструмента в автоматическом режиме для работы. Предварительно прежде чем исполнить данные G - команды как правило выключают калибровку инструмента станка.

G28 – возвращение позиции при старте в автоматическом режиме. Команда G28 служит для возвращения стартовой позиции станка. То есть быстрое позиционирование рабочего инструмента в позицию условного нуля станка. Возврат к условной нулевой позиции используется для условия параметров проверки и качества обработки материала в программе посередине обработки. Довольно часто команду G28 используют в конечном коде программы управления, для того чтобы по окончании программы, рабочий стол вернулся в позицию, для удобного извлечения обработанного материала.

Для перемещения в исходное положение используется условный кадр следующего вида:

```
G90 G28 X0.0 Y0.0 Z0.0
```

В кадре с G28 задаются оси с 0.0 параметрами, перемещение в позицию при старте осуществляется по трем осям. Но не всегда обязательно выполнять данную операцию. Иногда требуется перемещение только по одной из них. Например, для перемещения по осям Z в программном коде для обработки необходимо использовать данный шаг:

```
G90 G28 Z0.0
```

Пристальное внимание необходимо обратить на установленный в кадре код G90. Данная команда включает работу станка в относительных координатах. Иначе работает команда G28, она производит программирование некой промежуточной точки, куда будет перемещен инструмент, затем портал возвратится в начальную позицию. Действительно координаты станка, введенные в кадре программы, есть не что иное как координаты промежуточной точки. В данных примерах мы вводили в координаты промежуточной точки с нулевыми значениями. В программном кадре указана команда G90 являющаяся относительной координатой, станок будет перемещаться относительно настоящего положения по каждой оси на ноль миллиметров. Иными словами не будет никуда перемещаться. Вот поэтому при наличии управляющем коде кадра G90 G28 X0.0 Y0.0 Z0.0 портал вернется в начальную позицию без перемещения в промежуточную точку.

При наличии в программе обработки кадра G90 G28 X10.0 Z20.0, портал станка изначально поедет вверх и вправо, а потом переместиться в точку ноль. Команда G28 позволяет осуществить перемещение с ускорением, как и G00, в данном моменте оно возможно будет непрямолинейным. Иными словами портал может что-то «зацепить» . Программист с большим опытом изначально поднимает шпиндель вверх, потом перемещает портал в начальную позицию:

```
G90 G28 X0.0 Y0.0 Z30.0
```

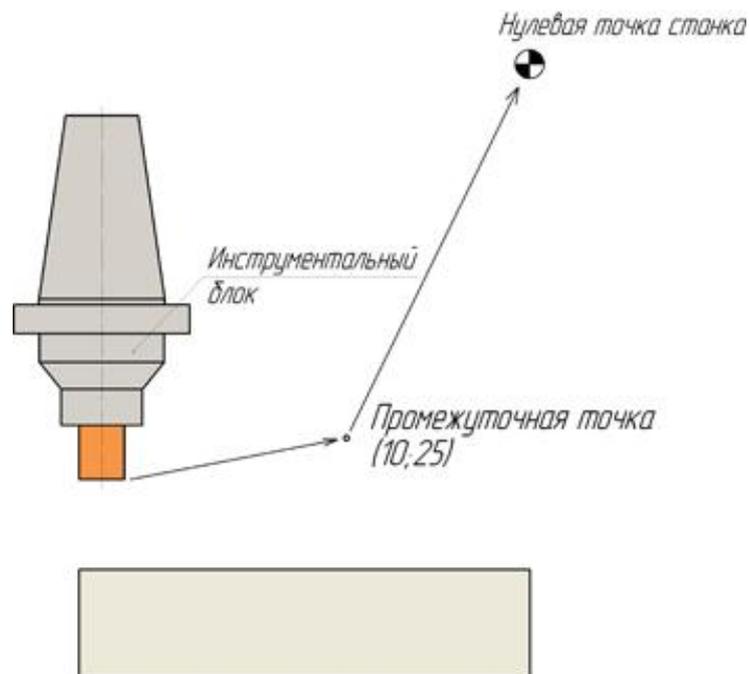


Рисунок 1.4.9 - Когда в программе есть команда G91 G28 X10 Y25, то портал в первую очередь уйдет в промежуточную точку, потом возвратится в начальную позицию.

Не стоит в кадре с G28 использовать команду с абсолютными координатами G91. Если программе имеется кадр G91 G28 X0 Y0 Z0, тогда есть возможность "наезда" рабочего портала на составные части станка или детали.

G30 – команда для возврата в положение смены инструмента. С помощью команды G30 осуществляется перемещение по оси Z к положению для замены инструмента в работе и выключается используемая инструментальная калибровка[11]. Для смены инструмента используется следующая команда подпрограммы:

G30 G90 Z0

Стоит заметить что, если в кадре программы вместо G91 присутствует команда G90, то шпиндель будет перемещаться к поверхности портала.

G31 – команда для пропуска с реакцией на импульс извне. В станках иногда используют команду для пропуска с реакцией на импульс извне. С

помощью немодальной команды G31, оператор выполняет интерполяцию по линии схожую с G01, смешанную с реализацией отклика импульс поступающий извне. Сигнал, поступающий от портала подается от нажатия на искомую кнопку панели управления станка, как пример клавиша Старт программы станка.

При отсутствии импульса пропуска, исполняемая программа работает, как если бы был задан код G01. Если станок принял поступившую команду, то программа выполняется и переходит к последующей операции.

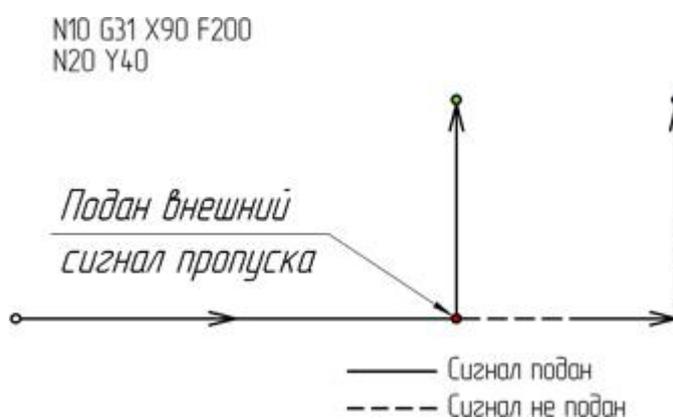


Рисунок 1.4.10 - Пропуск функцией с реакцией на внешний сигнал

G40 – команда отменяет коррекцию инструмента по радиусу. Коррекция инструмента автоматом по радиусу инструмента выключается путем подачи кодов D00 и G40. Команда G40 стоит в шаге с кодом прямолинейным перемещением с ускорением по контуру детали.

G1 G40 X90

G41 – корректировка радиуса, инструмент располагается слева от детали. Команда G41 применяется для активации коррекции радиуса инструмента в автоматическом режиме, находящегося от детали слева. Направление перемещения сверху вниз, при наблюдении портала сверху, от стороны «+Z» в положение «-Z».

G42 – коррекция радиуса , инструмент располагается с правой стороны от детали. Команда G42 применяется для активации автоматической корректировки радиуса рабочего инструмента, расположенного справа от детали. Направление смещения определяется, если смотреть сверху вниз, со стороны «+Z» в направлении «-Z».

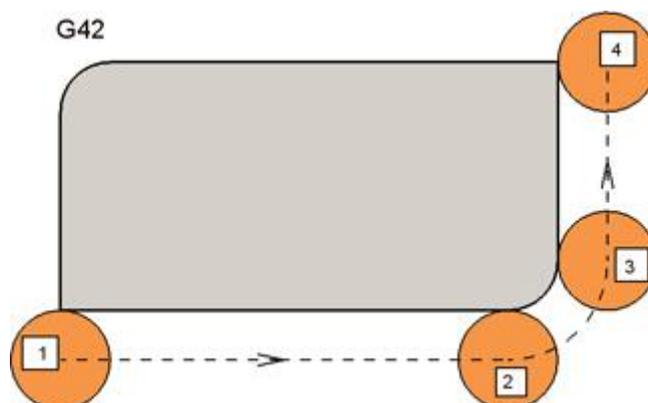


Рисунок 1.4.11 - Коррекция справа

G43 – корректировка длины инструмента. При выполнении управляющей программы начальная позиция рабочего инструмента определяется заданными координатами. Проблема заключается в том, что начальная позиция рабочего механизма не осуществляется. Обработка производится кромкой рабочего инструмента, которая находится на некотором расстоянии от начальной точки рабочего механизма. Для того чтобы в заданную координату приходил рабочий механизм, необходимо «показать» ЧПУ, на какое расстояние по оси Z нужно сместить эту стартовую точку.

Изменение длины инструмента происходит путем программирования команды G43 и H-слова данных. Обычно корректировка длины активируется совместно с ускоренным перемещением по оси Z.

Пример: G43 H01 Z90

G49 – отмена компенсации длины рабочего механизма. Компенсация длины рабочего механизма отменяется путем задания команды G49 или H00.

G50 – деактивация режима масштабирования. Код G50 предназначен для деактивации масштабирования G51.

G51 – активация режима масштабирования. В этом режиме программист изменяет коэффициент масштабирования для координатных осей станка. Режим включается при помощи модального кода G51 и отменяется G50.

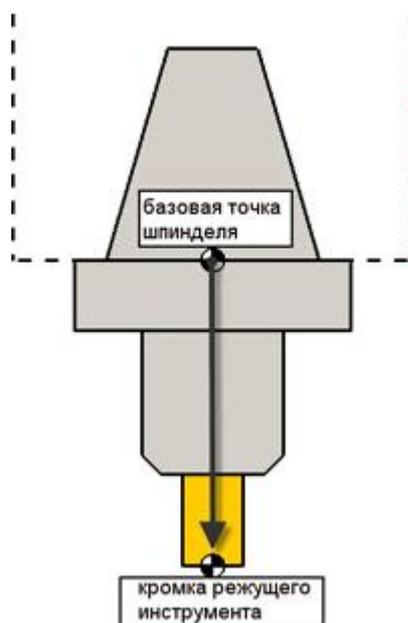


Рисунок 1.4.12 - Команда G43H\_ смещает базовую точку шпинделя к кромке режущего инструмента

Можно указать коэффициент масштаба для всех разом или для каждой оси. Если коэффициент масштабирования больше 1, то система координат увеличивается. Если же коэффициент масштабирования меньше 1, то система координат уменьшается.

Для единого изменения масштаба обычно используется следующий формат[13]:

G51 X\_Y\_Z\_P\_

где G51 – активирует режима масштабирования; X – координата по оси X для средней точки масштабирования; Y – координата по оси Y для средней

точки масштабирования;  $Z$  – координата по оси  $Z$  для средней точки масштабирования;  $P$  – коэффициент масштабирования для всех осей одновременно.

Возможно также зеркальное отображение с помощью отрицательных коэффициентов масштаба. Для независимого изменения масштаба обычно используется следующий формат:

G51 X Y Z I J K

где G51 – активирует режима масштаба;  $X$  – координата по оси  $X$  для средней точки масштабирования;  $Y$  – координата по оси  $Y$  для средней точки масштабирования;  $Z$  – координата по оси  $Z$  для средней точки масштабирования;  $I$  – коэффициент масштабирования для оси  $X$ ;  $J$  – коэффициент масштабирования для оси  $Y$ ;  $K$  – коэффициент масштабирования для оси  $Z$ .

В функции зеркального отображения комбинируются между собой независимое изменение масштабирования и возможность зеркального отображения заданных координат по одним или несколько осям.

### **Основная программа**

G90 G01 F90

M98 P101

G51 X5 Y5 I-1 J1 K1

M98 P101

G51 X5 Y5 I-1 J-1 K1

M98 P101

G51 X5 Y5 I1 J-1 K1

M98 P101

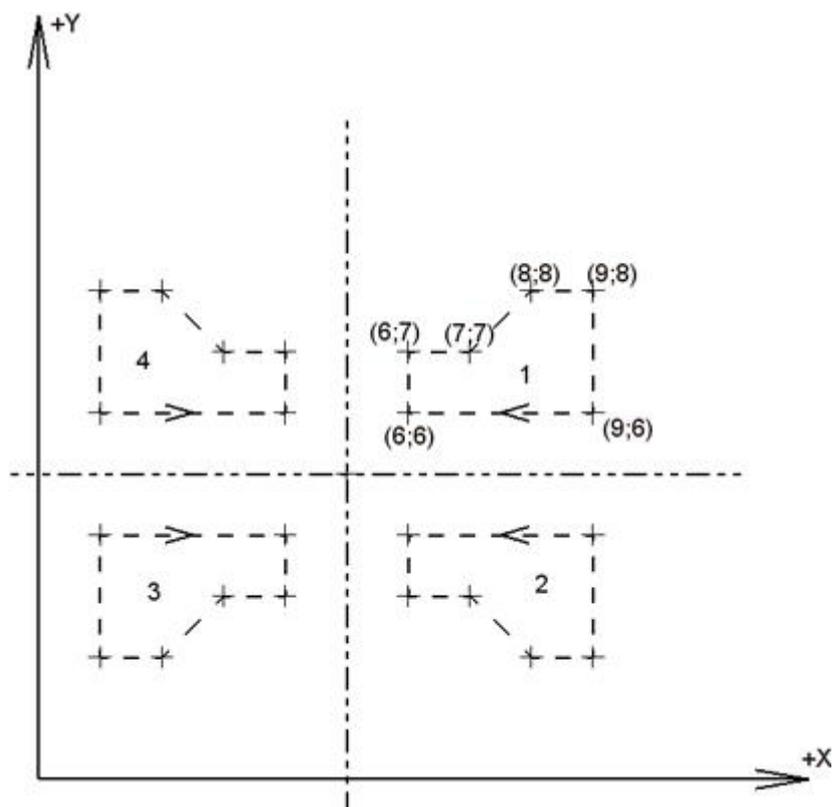


Рисунок 1.4.13 - Зеркальное отображение траектории

### Подпрограмма

O0101

G90 X6 Y6

Y7

X7

X8 Y8

X9

Y6

X6

M99

G52 – локальная система координат. ЧПУ позволяет задавать, кроме обычных рабочих систем позиционирования, еще и локальные системы координат. Код G52 применяется для определения подчиненной системы координат в диапазоне действующей рабочей системы (G54–G59).

Когда ЧПУ станка исполняет команду G52, то начало действующей рабочей системы координат смещается на значение, заданное при помощи слов данных X, Y и Z:

G52 X\_Y\_Z\_

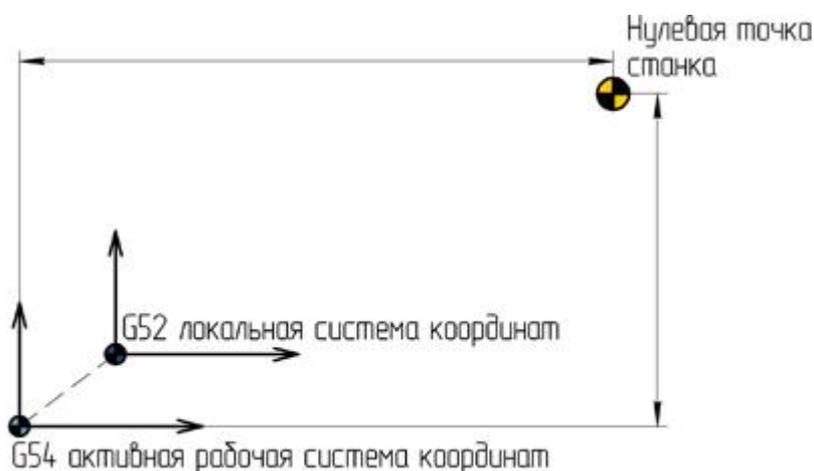


Рисунок 1.4.14 - Локальная система координат

Команда G52 деактивируется, если задается другая рабочая система координат G54–G59 или с помощью команды G52 X0 Y0 Z0

G54–G59 – стандартные действующие системы координат. При помощи команд G54, G55, G56, G57, G58 и G59 задается, в какой действующей системе координат будет выполняться обработка заготовки. Путем выбора различных координатных систем оператор имеет возможность при помощи одной и той же программы обрабатывать различные заготовки. Если была задействована одна из координатных систем G54–G59, то она действует до тех пор, пока не будет активна другая координатная система.

G60 – позиционирование в заданном направлении. При помощи команды G60 ко всем заданным позициям по каждой оси можно перемещаться из определенного направления. Благодаря этому получается возможность исключить ошибки координирования, которые могут возникать из-за неиспользованного хода в системах. Как правило, направление и величина позиционирования указываются параметрами ЧПУ.

G61 – режим очень точного позиционирования останова. Команда G61 используется для включения режима останова. Команда точного останова подробно описана в характеристике команды G09. Единственным отличием между кодами G61 и G09 заключается в том, что G09 является немодальной командой. Модальный код G61 остается включенной, пока не будет задана команда на изменение этого режима, например с помощью команды G63.

G64 – режим нарезания. Стандартный режим нарезания активируется кодом G64.

G65 – немодальный вызов микропрограммы. Код G65 позволяет выполнить микропрограммы, расположенные в памяти ЧПУ. Формат для немодального вызова микропрограммы выглядит таким образом:

G65 P\_L\_

где G65 – команда для вызова микропрограммы; P – номер микропрограммы; L – количество выполнений микропрограммы. Если L не указывается, то ЧПУ принимает, что  $L = 1$ .

G66 – модальный вызов микропрограммы. Команда G66 предназначена для вызова микропрограммы, как и команда G65. Единственная разница между двумя этими кодами заключается в том, что G66 является модальным кодом и микропрограммы выполняется при каждом перемещении, пока не будет использоваться команда G67. Формат для модальной активации микропрограммы:

G66 P\_L\_

где G66 – команда для циклов микропрограммы; P – номер микропрограммы; L – количество циклов микропрограммы.

Если L не указывается, то ЧПУ считает, что  $L = 1$ .

G67 – дезактивация модального вызова макропрограммы. При помощи указания кода G67 дезактивируется режим модального вызова макропрограммы G66.

G68 – вращение координатной системы. Модальная команда G68 позволяет выполнить поворот координатной сетки на заданный угол. Для реализации такого вращения требуется указать плоскость вращения, центр поворота и угол вращения. Плоскость поворота устанавливается при помощи кодов G17 (плоскость XY), G18 (плоскость XZ) и G19 (плоскость YZ). Если желаемая плоскость поворота уже активирована, то задание команд G17, G18 и G19 в шаге с G68 нет необходимости.

При действующей программы G90 центр вращения указывается абсолютными координатами относительно стартовой точки станка [14], если не активирована одна из стандартных рабочих систем позиционирования. Если выбрана одна из рабочих систем позиционирования G54–G59, то центр вращения задается относительно стартовой точки действующей рабочей системы позиционирования. В случае активной команды G91 центр поворота указывается относительно текущего позиционирования. Если же координаты цент вращения не будут указаны, то в качестве центра поворота будет принята текущая позиция.

Угол поворота указывается при помощи R-слова данных. Формат для команды разворота координат обычно следующий:

G17 G68 X\_Y\_R\_

G69 – отмена разворота координат. При помощи кода G68 отменяется режим поворота координат.

G73–G89 – постоянные циклы

<b>код</b>	<b>Описание</b>
80	Отмена постоянного цикла
81	Стандартный цикл сверления
82	Сверление с выдержкой
83	Цикл прерывистого сверления
73	Высокоскоростной прерывистого сверления цикл
84	Цикл нарезания резьбы
74	Цикл нарезания левой резьбы
85	Стандартный цикл растачивания

Таблица 1.4 - G кодов

G90 – активация режима абсолютного позиционирования. В этом режиме, абсолютного позиционирования G90 перемещения рабочих органов производятся относительно нулевой точки станка или относительно нулевой точки рабочей системы координат G54–G59. Код G90 является модальным и деактивируется при помощи кода относительного позиционирования G91.

G91 – активация режим относительного позиционирования. При помощи кода G91 включается режим относительного позиционирования. При инкрементном способе отсчета за стартовое положение каждый раз принимается положение рабочего органа, которое он занимал перед началом перемещения к следующей программируемой точке. Код G91 является модалным и деактивируется при помощи команды абсолютного позиционирования G90.

G92 – перемещение абсолютной системы координат. Бывают ситуации, когда у оператора станка возникает необходимость установить определенные переменные в регистрах абсолютной системы позиционирования для перемещения начальных точек в новое положение.

Код G92 используют для перемещения текущего положения начальной точки путем изменения значений в регистрах рабочих смещений. Когда ЧПУ выполнит программу G92, то значения в регистрах смещений изменятся и станут равными переменным, которые определены X-, Y- и Z-словами данных.

G92X Y Z

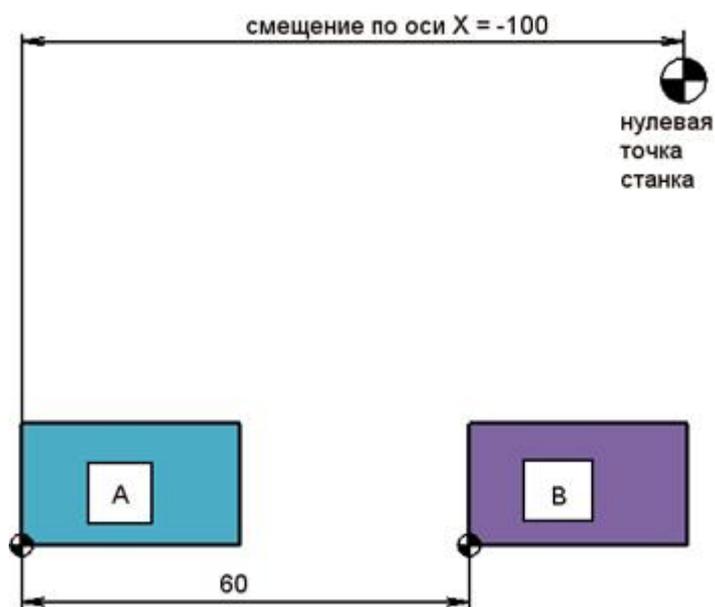


Рисунок 1.4.15 - С помощью G92 мы заменяем абсолютной позиции станка и смещаем нулевую точку

Сначала переместим рабочий механизм в известную нам начальную нулевую точку, а затем используем G92:

...

```
G00 X0 Y0
```

```
G92 X-60 Y0
```

...

Кадр G92 X-60 Y0 означает, что нынешнее текущее положение инструмента определено координатами (-60; 0). Благодаря этому, искомая начальная точка будет находиться на 60 мм правее текущего положения рабочего инструмента.

Существует иной метод для достижения этого же результата. Можно сначала переместить рабочий инструмент в позицию, которую надо принять новой нулевой точкой, и затем выполнить команду G92 X0 Y0.

...

```
G00 X60 Y0
```

```
G92 X0 Y0
```

...

Команда G92 сама по себе не выполняет осевых перемещений. Указанное при помощи G92 смещение координатной системы на большинстве станков может быть отменено возвратом в нулевую точку или выключением станка.

G94 – скорость движения. При помощи программы G94 указанная скорость подачи устанавливается в дюймах за 1 минуту или в миллиметрах за 1 минуту.

Если действует дюймовый режим G20, то скорость подачи F задается как подача в дюймах за 60 секунд. Если же активен метрический режим G21, то скорость подачи F принимается как подача в миллиметрах за 60 секунд.

G20 F10 – скорость перемещения 10 дюймов в минуту; G21 F10 – скорость перемещения 10 миллиметров в минуту.

Модальный код G94 остается включенным до тех пор, пока не будет запрограммирован код G95.

G95 – скорость подачи в дюймах/миллиметрах на вращение. За счет команды G95 указанная скорость подачи принимается в дюймах на 1 оборот шпинделя или в миллиметрах на 1 оборот шпинделя. Другими словами скорость подачи F синхронизируется со скоростью вращения шпинделя S. При равных значениях F скорость подачи будет увеличиваться при увеличении числа оборотов.

G20 F0.1 – скорость перемещения равна 0.1 дюйма на оборот; G21 F0.1 – скорость перемещения равна 0.1 миллиметра на оборот. Модальная программа G95 остается включенной до тех пор, пока не будет запрограммирован код G94.

G98 – возврат к исходной плоскости в программе. Исходная плоскость – это координата по оси Z (уровень), в которой находится инструмент перед активацией постоянного цикла. Команда G98 деактивируется при помощи команды G99.

G99 – возвращение к плоскости отвода в цикле. Если цикл работает совместно с кодом G99, то рабочий инструмент возвращается к плоскости отвода между всеми обрабатываемыми поверхностями. Плоскость отвода – это позиция по оси Z (уровень), с которой стартует на рабочей подаче и в которую возвращается рабочий инструмент после того, как он достиг дна обрабатываемой заготовки. Плоскость отвода обычно устанавливается в

кадре цикла при помощи R-адреса. Команда G99 отменяется с помощью программы G98.

### **Адреса/слова данных**

X является программой осевого перемещения. Всегда в практике, за основную ось берут ось X, относительно её будет максимальное перемещение инструмента станка. Эта ось находится строго перпендикулярно X к оси Z и стоит параллельно рабочему пространству обрабатываемой детали.

Положительное или отрицательное значение, входящее в состав этого слова, определяет конечную координату исполнительного органа станка вдоль оси X [15]. В кадре можно задать X только 1 раз. Если в 1 кадре будет несколько программ X, то ЧПУ выполняет команду начиная с последней, которая стоит ранее.

Код для примера:

G01 G90 X100 F20 –переход в систему координат по линии X = 20 с заданной скоростью 20 миллиметров в минуту

Когда X находится в одном кадре с программой выдержки G04, то оно задает время этой выдержки в секундах.

Код для примера:

G04 X11.0 – выполнить выдержку продолжительностью 11 секунд.

Y является программой осевого перемещения. Ось Y перпендикулярна для 2 осей X и Z. Положительное или отрицательное число, входящее в состав этого цикла данных, задает конечную позицию рабочего органа станка вдоль оси Y. В кадре можно запрограммировать Y только 1 раз. Если в строке будут указаны несколько команд Y, то ЧПУ выполняет команду начиная с последней, которая стоит ранее.

Код для примера:

G01 G90 Y10 F210 – переход в систему координат с линейной скоростью Y = 10 скорость равна 210 миллиметров в минуту.

Z является командой перемещения по оси. Для положительного изменения движения оси Z задают направление по вертикали для вывода рабочего инструмента из заготовки. Другими словами ось Z всегда связана со шпинделем станка. Положительная или отрицательная переменная, входящая в состав этого слова данных, задает конечную позицию выполняемого органа станка вдоль оси Z. В кадре можно задать Z только один раз. Если в кадре будут указаны несколько команд Z, то ЧПУ будет работать той командой, которая находится краю кадра программы.

Код для примера:

G01 G90 Z0.7 F300 –перемещение в системе координат по линии к  $Z = 0.7$  с заданной скоростью 300 миллиметров в минуту.

A, B, C являются программами иного перемещения. Под перемещением по кругу подразумевается угловое перемещение, поворот оси шпинделя станка или угловое перемещение (разворот) управляемого вращением стола.

поворотные перемещения инструмента обозначают латинскими буквами – A (вдоль оси X), B (вдоль оси Y) и C (вдоль оси Z). Для определения вращения осей в положительном направлении используется следующий код.

Код для примера:

G01 G90 C40 F300 – разворот портала на  $40^\circ$  со скоростью 300 миллиметров в минуту.

I, J, K используются для интерполяции по кругу и необходимы для обозначения относительных расстояний от точки начала дуги до центральной точки. Слово с параметром I принадлежит оси X, слово с параметром J – принадлежит оси Y, а слово с параметром K – принадлежит оси Z. При различном положении дуги значения принимают отрицательную или положительную форму.

R. При текущей круговой интерполяции (G02/G03) R показывает радиус, который соединяет начальную и конечную точки дуги.

Для большинства ЧПУ адрес R может являться командой на выполнение округления при текущей линейной интерполяции. Числовые переменные, входящие в состав R-слова данных, определяет радиус округления.

В постоянных циклах R определяет местоположение плоскости отвода. При использовании команды поворота координатной сетки R указывает угол вращения координатной плоскости.

R как правило применяется в циклах с помощью постоянных циклов и задает выдержку по времени на дне изделия. Числовое значение, входящее в состав R - слова данных, как правило, определяет время выдержки в 1 миллисекунду.

Когда R появляется в одном кадре с программой вызова подпрограммы M98, то оно указывает название искомой подпрограммы. В некоторых случаях данное слово данных может указывать на частоту вызова подпрограммы [16].

Код для примера:

M98 P10001 – вызов нужной программы O10001.

Q часто применяется в циклах, определяет глубину каждого рабочего хода инструмента.

В цикле Q задает сдвиг на расстояние инструмента обработки от стенки обработанного изделия для обеспечения точного удаления инструмента из зоны работы.

При помощи D задается значение коррекции на радиус рабочего инструмента. Коррекция радиуса рабочего инструмента активируется командами G41 и G42. При помощи программы D00 можно отменить действующую команду коррекции.

При помощи H выбирается значение компенсации длины рабочего инструмента. Компенсация длины рабочего инструмента, как правило активируется командой G43. При помощи команды H00 можно дезактивировать компенсацию длины действующего инструмента.

Для задания скорости подачи служит F-адрес. Если в 1 кадре будут запрограммированы множество скоростей подач, то ЧПУ будет использовать с последнюю из них. В случае задания функции F с командой G94 установленная скорость подачи инструмента будет принята в дюймах (G20) либо миллиметрах в одну минуту (G21). А в случае использования с G95 скорость подачи будет установлена в дюймах (G20) либо миллиметрах (G21), наоборот. Адрес - F есть модальная функция, по другому, внесенная скорость подачи инструмента держится постоянной до того времени, пока новая переменная совместно с F или не изменится включенный режим перемещений при помощи команды G00.

С помощью S применяется число оборотов. S-адрес является модальным, то есть установленное число оборотов остается постоянным до тех пор, пока не задано иной параметр значения вместе с S.

При помощи T указывается управление магазином инструментов. Числовое значение с T указывает номер инструмента, который необходимо переместить в позицию смены путем вращения инструментального магазина. Как правило T программируют в одном кадре с командой смены инструмента M06. В этом случае количество значений при T будет определять номер инструмента, который необходимо выбрать из магазина и установить в руку.

Код для примера:

T2 M06 – вызвать инструмент № 2.

С помощью N выполняется обозначение номеров УП. При выполнении кадрированного номера он может быть выбран в любой позиции кадра, но обычно его указывают в начале. Номер кадра не влияет на работу, а помогает программисту ориентироваться в содержании программы.

### **М-коды**

M00 – запрограммированная остановка. Когда ЧПУ применяет команду M00, то выполняется так называемый запрограммированный останов. Все осевые движения останавливаются и стартуют лишь затем, как на станке нажмут кнопку Циклический старт на панели управления. При

этом рабочий механизм продолжает работать и остальные функции остаются включенными. Если оператор включает кнопку Циклический старт, то исполнение цикла продолжается с кадра программы, следующего за кодом M00.

M01 – команда выборочной остановки. Код M01 служит для выборочной остановки. Работает он так же как и команда M00, но дает возможность оператору останавливать выполнение управляющей команды. Когда переключатель включен, то при отработке кадра программы M01 останавливается портал. Когда же переключатель выключен, то кадр M01 игнорируется и программа не прерывается.

M02 – конец цикла. Код M02 информирует о завершении программы.

M03 – прямое вращение. С помощью команды M03 активируется прямое (по час. стр.) вращение шпинделя с указанным числом оборотов (слово - S). Команда программы M03 остается выполняемой до того времени, пока она не выключится командами M05 или M04.

M04 – обратное вращение. С помощью команды M04 активируется обратное (против час. стр.) вращение с заданным числом оборотов (слово - S). Команда программы M04 действует до того времени, пока она не выключится с помощью команд M05 или M03.

M05 – останов. Команда M05 прерывает вращение шпинделя, но не выключает перемещения по осям.

M06 – смена инструмента. Посредством команды M06 инструмент, установленный в шпинделе, заменяется на инструмент, установленный в готовой кассете инструментов.

M19 – юстировка. С помощью кода M19 происходит юстировка по радиусу (разворот в определенном положении), для того чтобы выставить привод на позицию смены инструмента.

M20 – отмена юстировки. При помощи этой команды отменяется команда юстировки шпинделя M19.

M30 – конец программы. Код M30 информирует о завершении цикла.

M98 – вызов сторонней подпрограммы. Команда M98 предназначена для вызова сторонней подпрограммы. Вместе с данной командой вводится слово - P данных, оно указывает на номер искомой подпрограммы.

Код примера:

M98 P10001 – использовать подпрограмму O10001.

M99 – конец подпрограммы. С помощью команды M99 возврат к главной программе осуществляется по окончании подпрограммы, из которой было обращение к подпрограмме.

## **1.5 Анализ и обоснование выбора драйвера управления шаговым двигателем ЧПУ станка**

*Шаговый электродвигатель* (ШД) - это синхронный бесщеточный электродвигатель с несколькими обмотками, в котором ток, подаваемый в одну из обмоток статора, вызывает фиксацию ротора [17]. Для его работы требуется специальное электронное устройство – драйвер управления (ДУ) ШД. При выборе драйвера нужно учитывать параметры ШД и функции выполняемые драйвером.

*Сила тока.* Ток, который может обеспечить ДУ. Как правило, он регулируется в достаточно широких пределах, но ДУ нужно выбирать такой, который может выдавать ток, равный току фазы выбранного ШД. Желательно, чтобы максимальная сила тока ДУ была на 20 - 30% больше. Это дает запас на случай, если вы захотите получить больший момент от ШД. Производители периодически выпускают пары ШД/ДУ с оптимально выбранными характеристиками. Пример HY-TB3DV-M / NEMA23HS7430.

*Напряжение питания.* Его значение достаточно сильно влияет на момент силы и скорость переключения двигателя, а так же на вибрации, нагрев двигателя. Обычно максимальное напряжение питания ДУ примерно равно максимальному току  $I$ , умноженному на 8-10. Чем больше индуктивность двигателя - тем большее напряжение требуется для драйвера.

Существует эмпирическая формула  $U = 32\sqrt{L}$ , где L - индуктивность обмотки ШД. U - максимальное значение напряжения питания драйвера. Величина U, получаемая по этой формуле, приближительная, но она позволяет ориентироваться при выборе ДУ.

**Опторазвязка по входу.** Практически во всех ДУ, опторазвязка стоит обязательно т.к. пробой ключа может привести к мощному импульсу, который выведет из строя дорогостоящий ЧПУ- контроллер. Опторазвязка дополнительно уменьшает наведение помех от силовой части ДУ в управляющий контроллер.

**Функция подавления резонанса.** Резонанс ШД – явление, которое проявляется всегда и отличается только резонансной частотой. Она прежде всего зависит от момента инерции нагрузки, напряжения питания ДУ и установленной силы тока фазы ШД. При возникновении резонанса ШД начинает вибрировать и терять крутящий момент, вплоть до полной остановки вала. Для подавления резонанса используется микрошаг и – встроенные алгоритмы компенсации резонанса. Колеблющийся в резонансе ротор ШД порождает микроколебания ЭДС индукции в обмотках, и по их характеру и амплитуде ДУ определяет, есть ли резонанс и насколько он силен. В зависимости от полученных данных ДУ несколько смещает шаги двигателя во времени относительно друг друга – такая искусственная неравномерность компенсирует резонанс.

**Протокол управления.** Надо убедиться, что драйвер работает по нужному вам протоколу, а уровни входных сигналов совместимы с требуемыми Вам логическими уровнями. Большинство ДУ работает по протоколу STEP/DIR/ENABLE и совместимо с TTL уровнем сигналов 0..5 В.

**Наличие защитных функций.** Среди них защита от превышения питающего напряжения, тока обмоток (в т.ч. от короткого замыкания обмоток), от переплюсовки питающего напряжения, от неправильного подключения фаз ШД.

**ТВ6560** - основной драйвер от Toshiba, очень распространен. Тактовая частота импульсов равна 15 кГц. Максимальный ток 3А. Присутствуют делители шага 1/1, 1/2, 1/8, 1/16. Защита от превышения тока и перегрева. Цена 5-5,5\$ за одну микросхему.

**ТВ6600** - новая разработанная микросхема (конец 2012 года) от Toshiba. Тактовая частота импульсов до 200 кГц. Выходной ток в пике до 4,5А. Присутствуют делители шага 1/1, 1/2, 1/4, 1/8, 1/16. Реализовано снижения тока при простое в работе. Есть дополнительные защиты. Для питания логики, не используются 5 вольт. Цена от 14-15,5\$ за одну микросхему, это на порядок больше чем у ТВ6560.

Исходя из этих данных, выбираем драйвер на основе микросхем ТВ6560АНQ.

В данном станке мы используем шаговые двигатели марки NEMA23HS7430. Они обладают следующими характеристиками представленными ниже.

### **Параметры шагового двигателя NEMA23HS7430**

Основные параметры шагового двигателя представлены в таблице.

Количество фаз	2 фазы
Шаг поворота	$1.8^0 \pm 5\%$
Напряжение	3 В
Ток	3 А / 1 фаза
Сопротивление	1 Ом $\pm$ 10%, Ом/Фаза
Индуктивность	3.2 мГн $\pm$ 10% мГн/Фаза
Удерживающий момент	190 Н*см , Мин.
Встречное удерж. сопротивление	6 Н*см , Макс.

Таблица 1.5 - Таблица основных параметров шагового двигателя NEMA23HS7430.

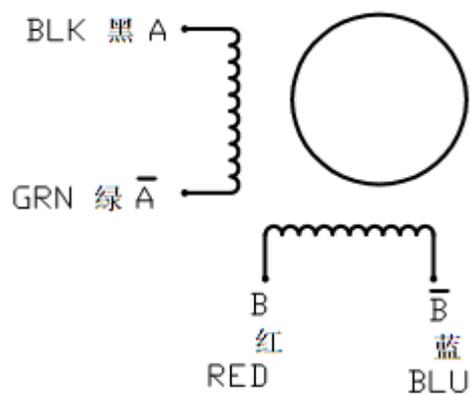


Рисунок 1.5.1 - Схема шагового двигателя NEMA23HS7430

На основе типового подключения была разработана принципиальная схема драйвера управления шаговым двигателем.

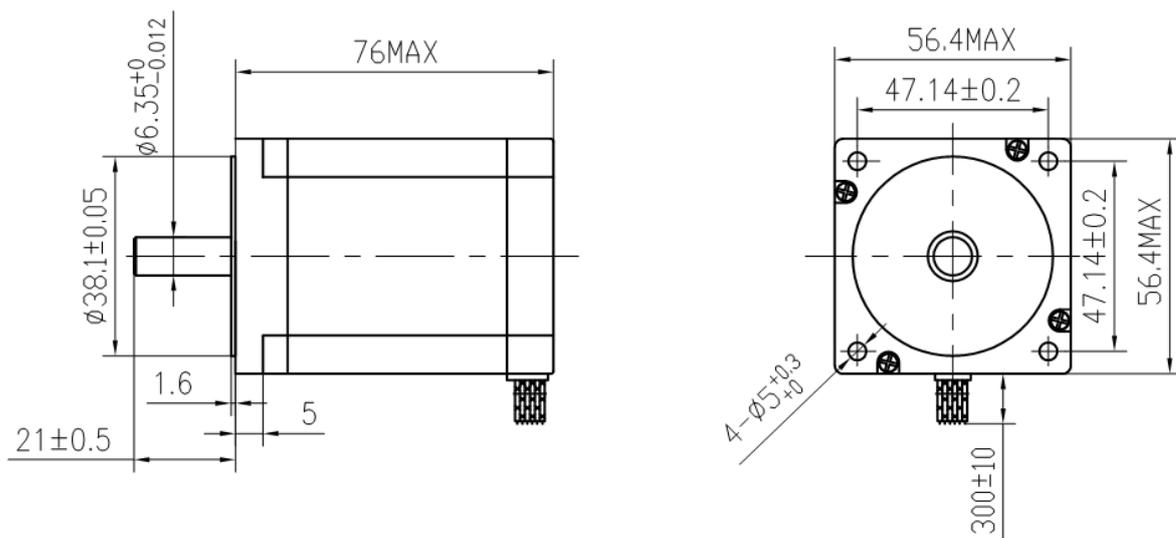


Рисунок 1.5.2 - Шаговый двигатель NEMA23HS7430

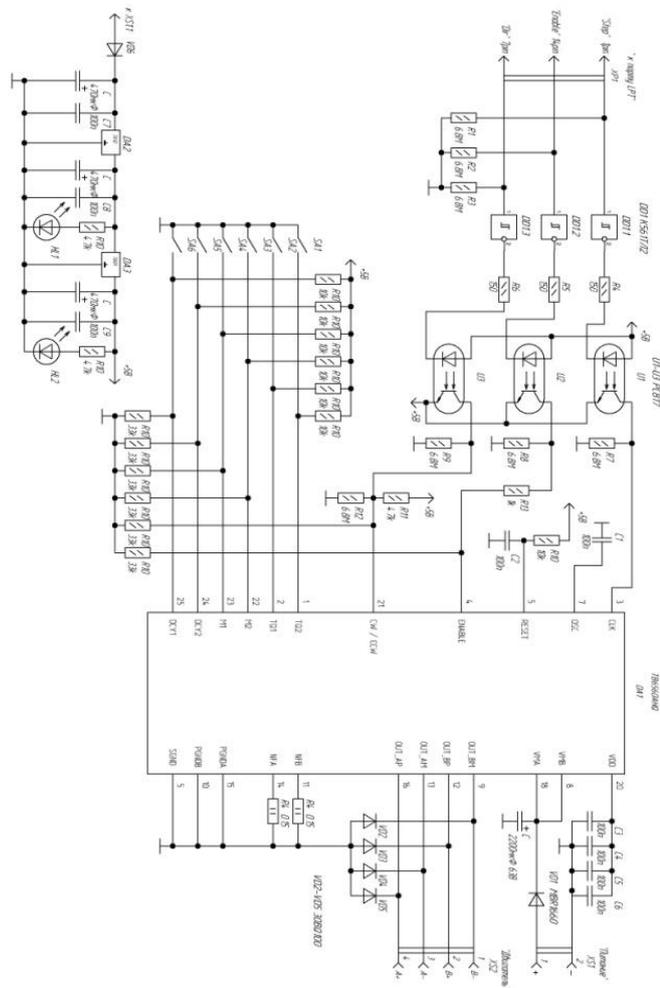


Схема электрическая принципиальная оси X

Рисунок 1.5.3 - Схема электрическая принципиальная оси X драйвера ТВ6560АНQ

## Драйвер управления шаговым двигателем ТВ6560АНQ

На рисунке показаны основные электрические характеристики драйвера управления шаговыми двигателями ТВ6560АНQ

Рисунок - электрические характеристики ( $T_a = 25^\circ\text{C}$ ,  $V_{DD} = 5\text{ V}$ ,  $V_M = 24\text{ V}$ )

Характеристики		Обозначения	Режим испытаний	Min	Typ.	Max	Unit
Входное напряжение	Максим.	$V_{IN(H)}$	M1, M2, CW/CCW, CLK,	2.0	—	$V_{DD}$	V
	Миним.	$V_{IN(L)}$	RESET, ENABLE, DCY1, DCY2, TQ1, TQ2	-0.2	—	0.8	
Входной гистерезис напряжения		$V_{inHys}$		—	400	—	mV
Входной ток		$I_{IN(H)}$	M1, M2, CW/CCW, CLK, RESET, ENABLE, DCY1, DCY2, TQ1, TQ2 $V_{IN} = 5.0\text{ V}$ Internal pull-down resistor	30	55	80	$\mu\text{A}$
		$I_{IN(L)}$	$V_{IN} = 0\text{ V}$	—	—	1	
Ток питания VDD		$I_{DD1}$	Outputs: Open, RESET: H, ENABLE: H (2, 1-2 phase excitation)	—	3	5	mA
		$I_{DD2}$	Outputs: Open, RESET: H, ENABLE: H (4W1 -2, 2W1-2 phase excitation)	—	3	5	
		$I_{DD3}$	RESET: L, ENABLE: L	—	2	5	
		$I_{DD4}$	RESET: H, ENABLE: L	—	2	5	
Ток питания $V_m$		$I_{M1}$	RESET: H/L, ENABLE: L	—	0.5	1	mA
		$I_{M2}$	RESET: H/L, ENABLE: H	—	0.7	2	
Дифференциальное напряжение канал-канал		$\Delta V_o$	B/A, $C_{osc} = 330\text{ pF}$	-5	—	5	%
ИЗМЕНЕНИЕ НАПРЯЖЕНИЯ В СООТВЕТСТВИИ С КРУТЯЩИМ МОМЕНТОМ $V_{NF}$		$V_{NFHH}$	TQ1 = H, TQ2 = H	10	20	30	%
		$V_{NFHL}$	TQ1 = L, TQ2 = H	45	50	55	
		$V_{NFLH}$	TQ1 = H, TQ2 = L	70	75	80	
		$V_{NFLL}$	TQ1 = L, TQ2 = L	—	—	100	
Минимальная частота тактовых импульсов		$t_w(\text{CLK})$	$C_{osc} = 330\text{ pF}$	30	—	—	$\mu\text{S}$
ОСТАТОЧНОЕ НАПРЯЖЕНИЕ $M_o$ ВЫХОДНОЕ		$V_{OL M_o}$	$I_{OL} = 1\text{ mA}$	—	—	0.5	V
Защита выходного напряжения		$V_{OL Protect}$	$I_{OL} = 1\text{ mA}$	—	—	0.5	V
Порог TSD		TSD	—	—	170	—	$^\circ\text{C}$
Гистерезис TSD		TSDhys	—	—	20	—	$^\circ\text{C}$
Частота осциллятора		$f_{osc}$	$C_{osc} = 330\text{ pF}$	60	130	200	kHz

Таблица 1.5.2 - Электрические характеристики ТВ6560АНQ

Таблица абсолютные максимальные характеристики ( $T_a=25^{\circ}\text{C}$ )

Характеристики		Обозначения	Параметры	Unit
Напряжение питания		$V_{DD}$	6	V
		$V_{M_{A/B}}$	40	
Выходной ток (1 фаза)	Пиковая	TB6560AHQ	3.5	A
		TB6560AFG	2.5	
МО ток утечки		$I_{(MO)}$	1	mA
Ток защиты		$I_{(Protect)}$	1	mA
Входное напряжение		$V_{IN}$	$V_{DD}$	V
Мощность рассеивания	TB6560AHQ	$P_D$	5 (Примечание 1)	W
			43 (Примечание 2)	
	TB6560AFG		1.7 (Примечание 3)	
	4.2 (Примечание 4)			
Рабочая температура		$T_{opg}$	-30 to 85	$^{\circ}\text{C}$
Максимальная рабочая температура		$T_{stg}$	-55 to 150	$^{\circ}\text{C}$

Таблица 1.5.3 - Абсолютные максимальные характеристики

Примечание 1 :  $T_a = 25^{\circ}\text{C}$  , без радиатора .

Примечание 2 :  $T_a = 25^{\circ}\text{C}$  , с бесконечным радиатором ( NZIP25 ) .

Примечание 3 :  $T_a = 25^{\circ}\text{C}$  , с припаянными проводами .

Примечание 4 :  $T_a = 25^{\circ}\text{C}$  , при установке на плате ( 4 - слойная плата).

Функциональная схема драйвера ТВ 6560АНQ представлена на рисунке

Рисунок - функциональная схема драйвера ТВ 6560АНQ

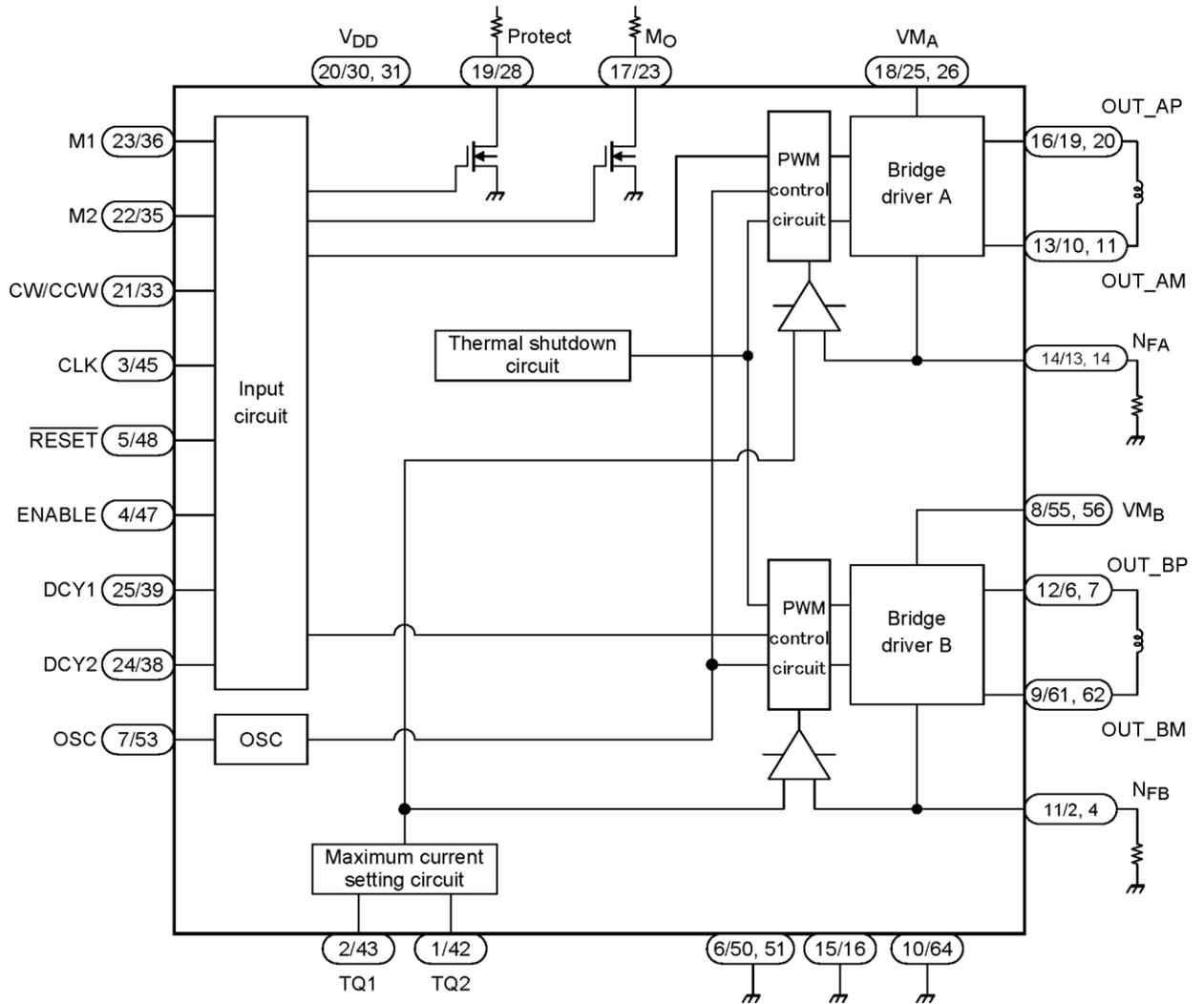


Рисунок 1.5.4 - Функциональная схема драйвера ТВ 6560АНQ

## Описание функций драйвера ТВ 6560АНQ

### Режим возбуждения

Режим возбуждения может быть выбран из следующих четырех режимов с использованием входов M1 и M2. ( Режим 2 - фазы возбуждения выбирается по умолчанию , так как M1 и M2 имеют внутренние подтягивающие резисторы. ) На рисунке представлена таблица переключений.

Рисунок - настройка режимов работы шаговых двигателей

Входы		Параметры двигателя
M2	M1	
L	L	2-фазы
L	H	1-2-фазы
H	L	4W1 -2-фазы
H	H	2W1-2-фазы

Таблица 1.5.4 - Настройка режимов работы шаговых двигателей

### Связь между входами и режимах на выходе

Когда уровень на входе Enable низкий, выходы выключены. Когда уровень на входе Reset низкий, выходы переключаются в начальный режим , как показано в таблице . В этом режиме, состояния входных сигналов CLK и CW / CCW игнорируются.

Входные сигналы				Выходной режим
CLK	CW/CCW	RESET	ENABLE	
	L	H	H	CW
	H	H	H	CCW
X	X	L	H	Режим инициализации
X	X	X	L	Z

Таблица 1.5.5 - Состояние входов управления ТВ6560АНQ

## Режим инициализации

При сбросе , фазовые токи в каждом режиме возбуждения будут следующими . В это время , выход  $M_O$  переходит на низкий уровень ( соединение с открытым стоком ) .

Режим возбуждения	А-Фазовый ток	В-Фазовый ток
2-фазы	100 %	-100 %
1 -2-фазы	100 %	0 %
2W1 -2-фазы	100 %	0 %
4W1 -2-фазы	100 %	0 %

Таблица 1.5.6 - Фазовые токи в режиме инициализации ТВ 6560АНQ

## Настройка режима спада тока ШИМ

Она занимает около четырех OSC циклов для спада тока в ШИМ режиме . Режим спада 25 % создается за счет индукции спада в течение последнего цикла в режиме быстрого спада ; режим Распад 50 % создается за счет индукции спада в течение последних двух циклов в режиме быстрого спада ; и режим спад 100 % создается за счет индукции спада в течение всех четырех циклов в режиме быстрого спада .

Так как контакты DCY1 и DCY2 имеют внутренние подтягивающие резисторы, нормальный режим выбран , когда DCY1 и DCY2 не будут подключены . В таблице показаны режимы настройки тока спада.

DCY2	DCY1	Настройки спада тока
L	L	Нормальный 0 %
L	H	25 % задержка
H	L	50 % задержка
H	H	100 % задержка

Таблица 1.5.7 - Настройки спада тока драйвера ТВ6560АНQ

## Настройки крутящего момента ( Уровень тока )

Отношение тока, необходимого для реальных операций с задаваемыми параметрами, регулируется с помощью внешнего резистора и могут быть выбраны следующим образом. Режим слабого возбуждения должен быть выбран для установки чрезвычайно низкого крутящего момента , когда двигатель находится в фиксированном положении .

Так как контакты TQ2 и TQ1 имеют подтягивающие резисторы , установка крутящего момента 100 % происходит, когда TQ2 и TQ1 не подключены. Параметры представлены в таблице.

TQ2	TQ1	Выходной ток
L	L	100 %
L	H	75 %
H	L	50 %
H	H	20 % (Низкий крутящий момент)

Таблица 1.5.8 - Выходной ток драйвера ТВ6560АНQ

### Защита выходных выводов M<sub>O</sub>

Выходы выполнены с открытым стоком . При использовании к этим выводам должен быть добавлен внешний нагрузочный резистор. Если цепь TSD активируется , защита приводится отключает выходную цепь . Когда драйвер вернется в исходное состояние , M<sub>O</sub> снова будет разблокирован .

Состояние вывода	Защита	M <sub>O</sub>
Низкий	Отключение по температуре	Начальное состояние
Высокий -Z	Нормальный режим	Кроме начального состояния

Таблица 1.5.9 - Защитные блокировки ТВ6560АНQ

## 1.6 Анализ и обоснование выбора микроконтроллеров для управления ЧПУ станками

На большинства, производствах для работы ЧПУ станков используется в качестве системы управления промышленный компьютер (ПК) [18], который выполняет роль считывания параметров обработки и траекторий движения рабочего механизма из написанной на G-коде программы и передаче их станку. Однако, данный ПК достаточно дорогостоящие и занимаю приличное место, если выполняемая программа не подразумевает частое изменение и редактирование параметров и траекторий то можно использовать достаточно дешевый и компактный пульт управления на микроконтроллере, который будет считывать из памяти программу и преобразовывать её в машинные команды для изготовления конечной продукции. Для выбора нужного микроконтроллера который отвечал бы всем требованиям ознакомимся с причиной появления, историей создания и модельным рядом который представлен сейчас на рынке.

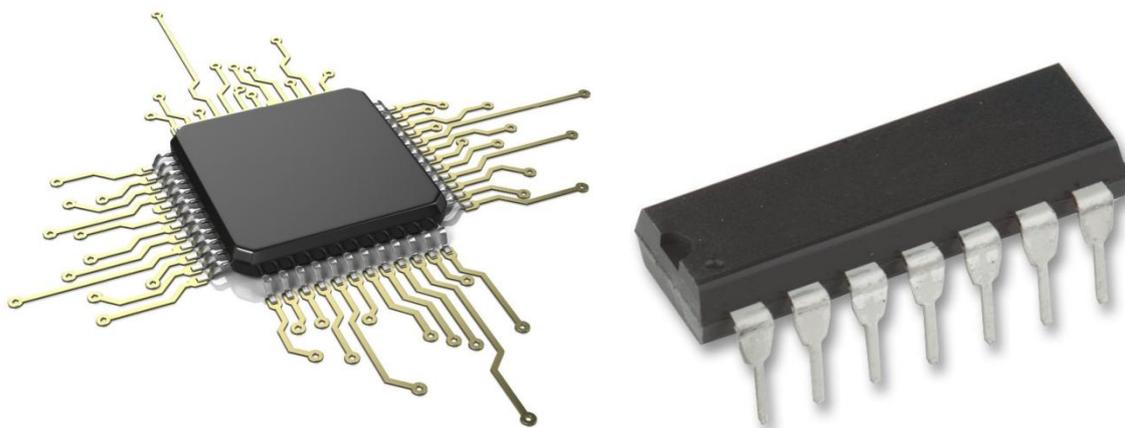


Рис. 17. Микроконтроллеры

Одна из основная причин стимулирующая появление микроконтроллеров - внедрение средств вычислительной техники во все сферы человеческой деятельности, в том числе и в промышленной. Это повлекло миниатюризации и снижения стоимости изделий. Прогресс же в

технологии создания микросхем в значительной степени способствовал этому.

В 1967 г. фирма Texas Instruments выпустила 1 калькулятор на интегральной микросхеме, который и запустил процесс миниатюризации средств вычислительной техники. Однако микросхемы для калькуляторов создавались применительно к требованиям каждого конкретного заказчика. Эти процессоры для калькуляторов были маломощными и не отвечали требованиям, предъявляемым к вычислительным мощностям.

Работая над заказом одной из японских фирм над разработкой программируемых калькуляторов, специалист фирм Intel предложил новую концепцию проектирования микропроцессоров, которая заключалась в создании микропроцессоров общего назначения, способных выполнять абсолютно любые арифметико-логические операции [19]. В 1969 г. фирмой Intel был представлен первая в мире специализированный интегральный чип Intel 4004 промышленный серийный выпуск с 1971 г., который по сути, являлся 4-разрядным программируемым микропроцессором. Микропроцессор Intel 4004 содержал примерно 2250 транзисторов и мог выполнять примерно 60 тыс. операций в секунду. Производительность Intel 4004 была слишком низкой, чтобы он мог в полной объеме выполнять функции процессора. Однако он мог служить основным микропроцессором для калькуляторов. Интересно факт, что название фирмы Intel происходит от слов "Integrated Electronics" (интегральная электроника). Возможно, смысловое содержание названия Intel играет не малую роль в том, что фирма с момента своего создания (1968 г.) и по настоящее время (сегодня в фирме Intel работают около 65 тыс. сотрудников) занимает лидирующее положение среди мировых производителей микросхем.

В 1976 г. создан первый 8-разрядный микроконтроллер, на кристалле которого были интегрированы основные элементы микропроцессорной управляющей системы: процессор, память типа ROM и RAM, порты ввода/вывода и таймеры.

В 1978г. появился 16-разрядный микропроцессор Intel 8086 (отечественный аналог К1810ВМ86) - первый микропроцессор массового использования. С серийного выпуска Intel 8086 началось масштабное проникновение персональных компьютеров во все сферы деятельности. Созданию микропроцессора Intel 8086 предшествовал выпуск 8-разрядных микропроцессоров Intel 8008 (1972 г.) и Intel 8080 (1974 г.).

Создание фирмой Texas Instruments в 1982 г. первого однокристального программируемого цифрового сигнального процессора (DSP) TMS32010. В течение времени из-за сложности алгоритмов и специфики математических операций, используемых при создании интеллектуальных приводов электрических двигателей, архитектура сигнальных процессоров используется в качестве основных для современных универсальных программируемых DSP-контроллеров, широко применяемых в системах управления электродвигателями всех типов.

Структурная схема любой электронно-вычислительной машины содержит следующие блоки: процессор, состоящий из арифметико-логического устройства (ALU), схем управления и регистров; память; периферийные устройства ввода/вывода данных.

Первоначально блоки, входящие в ЭВМ, создавались на базе стандартных дискретных логических микросхем, которые выполняли сравнительно простые функции.

Поэтому сами электронно-вычислительной машины имели большие габариты (например, суперЭВМ Сгау-1 состояла из 300 тыс. микросхем и занимала объем порядка трех кубических метров).

Успехи интегральной технологии привели к появлению больших и сверхбольших интегральных схем (БИС и СБИС) с размещением до 10 и 100 тысяч, а в настоящее время - 10 миллионов транзисторов на одном кристалле. Высокая степень внедрения БИС и СБИС позволила в одной микросхеме реализовать отдельные блоки ЭВМ, к примеру, процессор.

В микросхемах первых микропроцессоров (например, Intel 8080) были реализованы только сам процессор и дополнительные устройства, осуществляющие управление обменом данными с внешней памятью и устройствами ввода/вывода данных.

Однако, кроме устройств, входящих в состав процессора, на кристалле СБИС (БИС) могут быть выполнены память для хранения программ (ROM), данных или промежуточных результатов (RAM), периферийные устройства ввода/вывода. Такие СБИС (БИС) относятся к классу однокристалльных микро ЭВМ. Одной из 1 серийно выпускаемых микросхем однокристалльных микро ЭВМ стала Intel 8048 (18048). Однокристалльные микро ЭВМ начали активно использоваться там, где была потребность в несложной цифровой обработке данных: бытовых приборах, простых системах управления, контроля и т. п. Кроме однокристалльных микро ЭВМ для цифровой обработки в подобных системах можно использовать и заказные, специализированные интегральные микросхемы (Application Specific Integrated Circuit - ASIC).

Идея интеграции на одном кристалле совместно с процессором и памятью большого количества стандартных устройств, всевозможных назначения воплотилась в появление микроконтроллеров. Одним из 1 серийно выпускаемых микроконтроллеров можно считать Intel 8051. Вскоре микроконтроллер 8051 завоевал популярность. В настоящее время микроконтроллеры с набором команд 8051 выпускаются 10 фирм-производителей Analog Devices, Atmel, Dallas, Semiconductor, Oki, Philips, Infineon Technologies, Silicon Storage Technologies, Temic и многими другими.

В отличие от универсальных микропроцессоров, предназначенных в основном для математической обработки данных, микроконтроллеры имеют расширенный набор встроенных периферийных устройств. Это могут быть доп. блоки памяти типа RAM, ROM, EPROM, EEPROM или FLASH. Периферийные устройства всевозможного назначения: универсальные таймеры и таймеры специального назначения; "сторожевые" таймеры;

контроллеры внешних интерфейсов (UART, USART, SPI, SCI, PC, j 1850, USB- или CAN-шины) и жидкокристаллических дисплеев; монитор источников питания; аналоговые и цифровые компаратора; схему перезапуска; аналого-цифровые (АЦП) и цифро-аналоговые (ЦАП) преобразователи и др. Таким образом, микроконтроллеры содержат все периферийные устройства, нужные для создания законченных встроенных систем управления, контроля и что немаловажно для дальнейшего понимания, стандартные устройства, которые в случае использования микропроцессора в системе выполнялись бы на базе дополнительных внешних по отношению к микропроцессору микросхем. В микроконтроллерах по крайней мере, большая часть периферийных устройств выполнены на одном кристалле с процессором, что придает системам на базе микроконтроллеров значительно большую гибкость и универсальность. Примером подобных высокоинтегрированных микроконтроллеров могут служить микроконвертеры ADcU812/814/816/824/834 (Analog Devices), H8/300L (Hitachi), 51XA-G49 и 51XA-G3 (Philips), MC68HC05/08/ 11/12/16, микроконтроллеры семейств AT89/AT 90 (AVR)/AT tiny/AT mega (Atmel), C16x (Infineon), H8/300 и (Motorola), PICmicro (Microchip), MSP430F (Texas instruments) и многие другие.

### **Что такое микроконтроллер**

Прежде всего, разберемся с самим понятием «микроконтроллер». Микроконтроллер можно определить как миниатюрный компьютер на базе одного-единственного чипа, включающий, помимо процессора ряд вспомогательных элементов, таких, как ОЗУ, ППЗУ, таймер, и т.д. Микроконтроллер предназначен для выполнения каких-либо заранее определенных заданий.

Проще всего сравнивать микроконтроллер с ПК. Как и ПК, микроконтроллер имеет процессор, ОЗУ и постоянную память [20]. Однако, в

отличие от персонального компьютера, все эти элементы расположены на одном чипе.

ПК создан для того, чтобы выполнять громоздкие задачи общего назначения. Например, вы можете использовать компьютер, для набора текста, написания программ, хранения и запуска видео файлов, путешествия в Интернет, и.т.д. Микроконтроллеры предназначены для выполнения конкретных заданий, например, коммутации кондиционера, когда температура в помещении опускается ниже определенного значения, или наоборот выше.

Существует несколько популярных семейств микроконтроллеров, которые используются для всевозможно различных целей. Наиболее распространенными из них являются семейства микроконтроллеров 8051 [21], PIC и AVR. И последние в настоящее время набирают особую популярность на счет проекта Arduino «Микроконтроллеры это просто». В следствие чего документации на них достаточно много и существует большое сообщество по поддержке начинающих разработчиков и как следствие популяризации цены на них значительно меньше чем у конкурентов.

**Семейство микроконтроллеров AVR было основано в 1996 г. компанией Atmel, а разработчиками архитектуры являются Alf-Egil Bogen и Vegard Wollan.** Отсюда и происходит название семейства – от первых букв имен разработчиков – А и V, и первой буквы аббревиатуры RISC – типа архитектуры, на которой основана архитектура микроконтроллеров. Также это сокращение часто расшифровывают как Advanced Virtual RISC (модернизированный эффективный RISC).

Одним из первым микроконтроллером в серии был AT90S8515, однако 1 микроконтроллером, выпущенным на рынок, стал AT90S1200. Это произошло в 1997 г.

На сегодняшний момент, на рынке доступны 3 линейки микроконтроллеров:

- TinyAVR – малый объем памяти, малые размеры, подходит для самых элементарных задач.

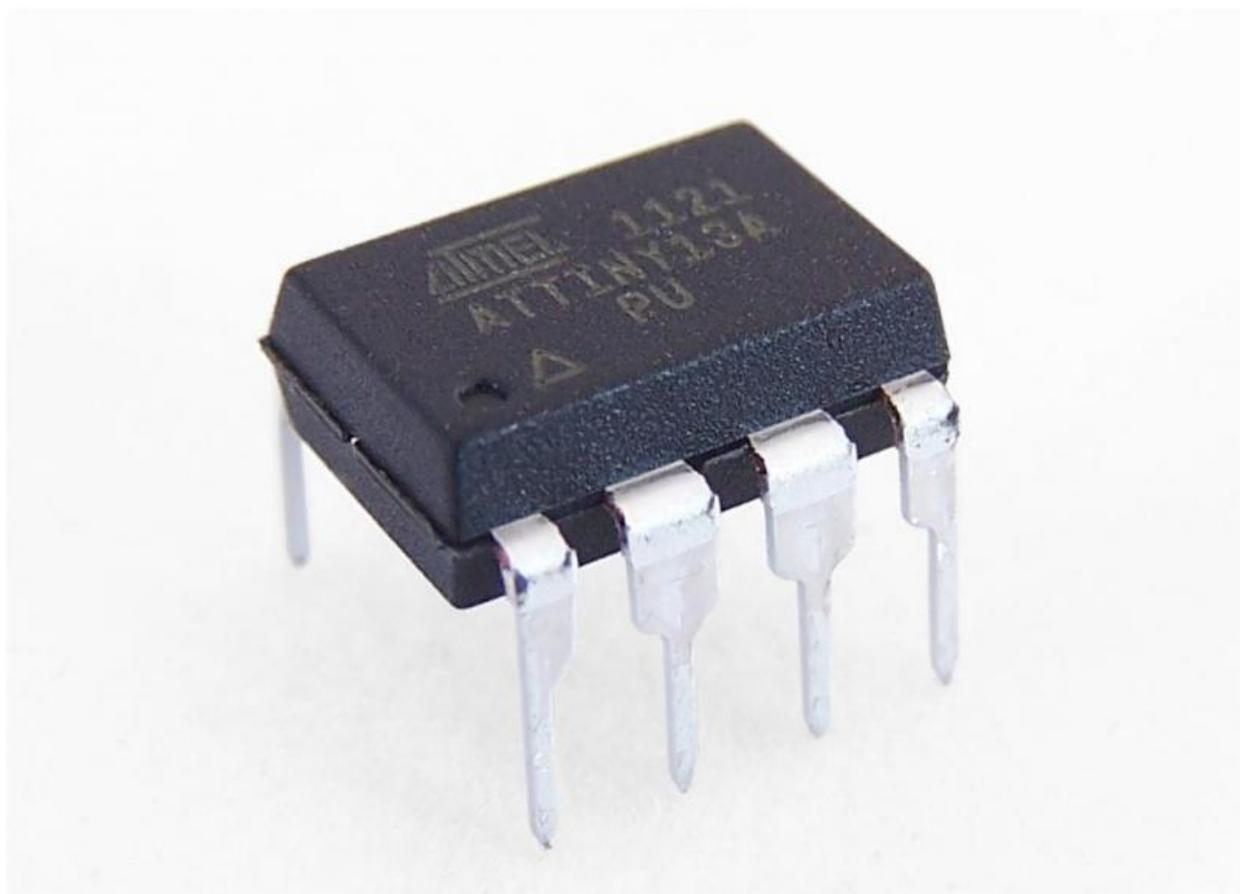


Рисунок 1.6.1 - Внешний вид микроконтроллера TinyAVR

- MegaAVR – наиболее популярная линейка, имеющая достаточный объем встроенной памяти (до 256 КБ), множество дополнительной периферии и служит для задач средней и высокой сложности.

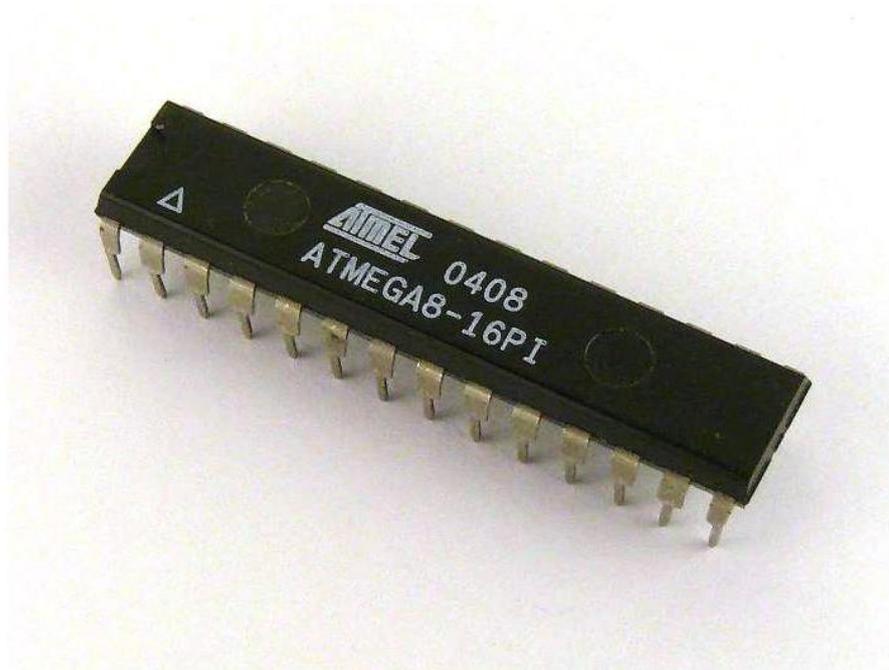


Рисунок 1.6.2 - Внешний вид микроконтроллера MegaAVR

- XmegaAVR – используется в продвинутых коммерческих задачах, требующих большого объема памяти и высокой производительности.



Рисунок 1.6.3 - Пример микроконтроллера XmegaAVR

Сравнительные характеристики различных линеек:

Название серии	Число контактов	Объем флэш-памяти	Особенность
TinyAVR	6-32	0,5 – 8 КБ	Небольшой размер
MegaAVR	28-100	4-256 КБ	Периферийные устройства
XmegaAVR	44-100	16-384 КБ	Система прерываний, поддержка DMA

Таблица 1.6.1 - Анализ технических характеристик микроконтроллеров

### **Особенности семейства**

Прежде всего, микроконтроллеры этой серии являются высокоскоростными. Многие инструкции процессор микроконтроллера выполняет за один цикл. Микроконтроллеры AVR примерно в 4 раза быстрее, чем PIC. Кроме того, они потребляют значительно меньше энергии и могут работать в нескольких режимах экономии энергии.

Многие контроллеры AVR являются 8-разрядными, хотя есть и 32-разрядная разновидность контроллеров AVR32. Кроме того, как уже говорилось выше, AVR принадлежат к типу RISC-микроконтроллеров. Архитектура RISC (Complex Instruction Set Computers) означает, что набор инструкций, которые может выполнять процессор устройства, является урезанными, но, в то же время, данная архитектура дает преимущество в быстродействии. Аналогом архитектуры RISC является архитектура CISC [22] (Complex Instruction Set Computers).

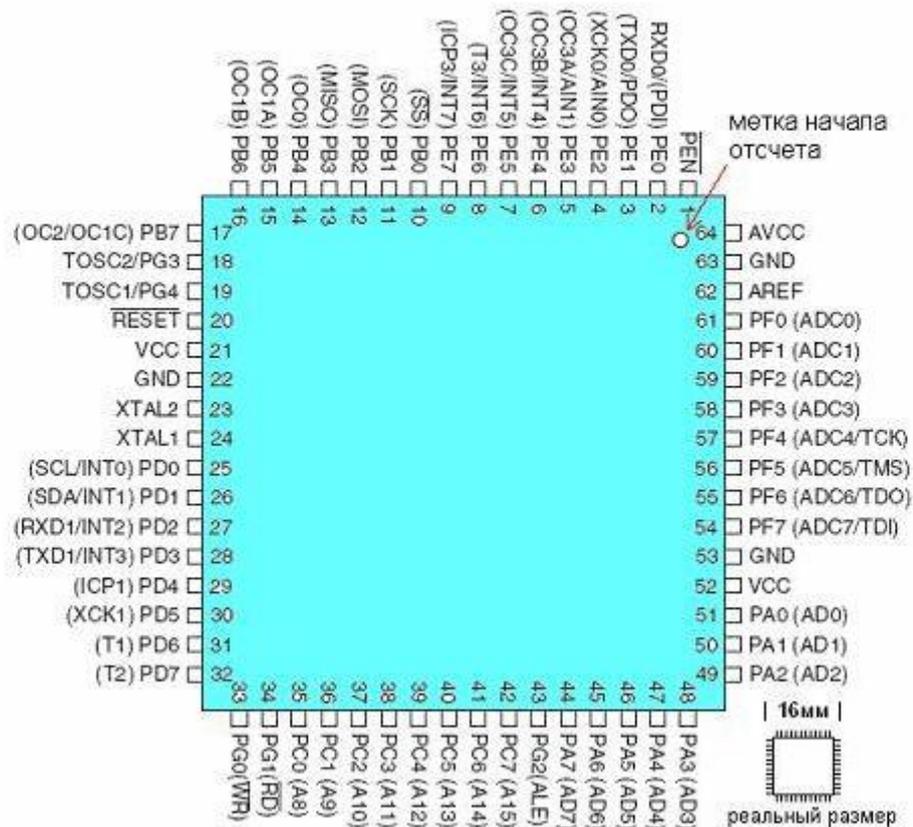


Рисунок 1.6.4 - Разрядная разновидность контроллеров AVR32

8-битность контроллера подразумевает, что он способен передавать и принимать 8-битные данные. Предоставленные регистры ввода/вывода также являются 8-битными.

Архитектура контроллера базируется на регистрах [23]. Это означает, что для хранения исходных значений операции и результата в контроллере применяются регистры/

Процессор контроллера получает данные из двух входных регистров, выполняет логическую операцию и сохраняет результат в выходном регистре. Все это занимает 1 исполняемый цикл.

### Архитектура контроллера

Всего контроллер AVR имеет 32 8-битных регистра общего применения. В течение цикла процессор берет данные из двух регистров и

использует их в арифметико-логическое устройство (АЛУ), которое производит вычисление над данными и помещает их в случайный регистр. АЛУ может выполнять как арифметические, так и логические действия над регистрами. Также АЛУ может выполнять и действия с одним регистром. При этом контроллер не имеет операнда-аккумулятора, в отличие от контроллеров семейства 8051 – для операций могут использоваться любые операнды, и результат операции также может быть помещен в любой операнд.

Микроконтроллер соответствует Гарвардской вычислительной архитектуре, согласно которой компьютер имеет независимую память для программ и данных. Поэтому в то время, пока выполняется одна операция, происходит предварительное извлечение из памяти следующей операции.

Котроллер способен выполнять одну операцию за цикл. Из этого следует, что если тактовая частота микроконтроллера составляет 1 МГц, то его производительность составит 1 млн. операция в секунду. Чем выше тактовая частота контроллера, тем выше будет его производительность. Однако при выборе тактовой частоты контроллера следует соблюдать компромисс между его скоростью и энергопотреблением.

Помимо флэш-памяти и процессора контроллер [24] имеет такие периферии, как порты ввода-вывода, аналого-цифровой преобразователь, таймеры, коммуникационные интерфейсы – I2C, SPI и последовательный порт UART. Все эти периферии могут контролироваться на программном уровне.

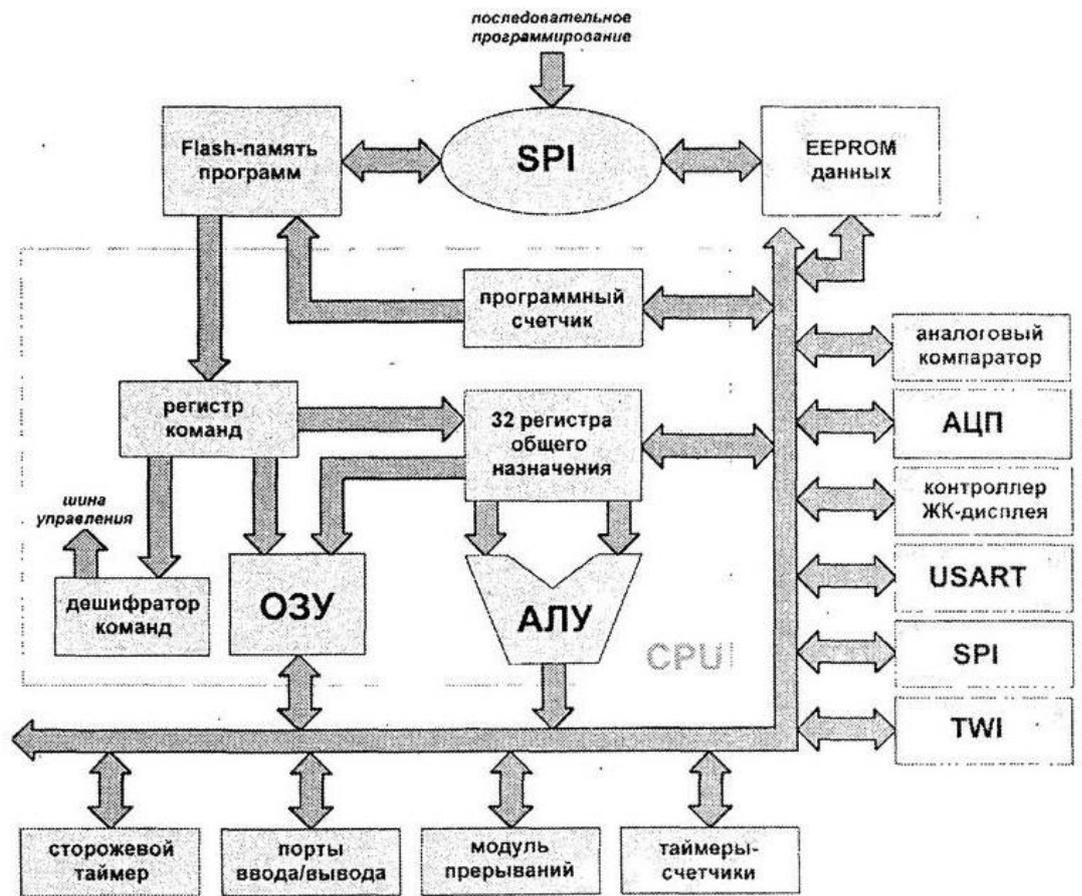


Рисунок 1.6.5 - Типовая архитектура микроконтроллеров AVR

## Программы для микроконтроллера

Как уже упоминалось выше, микроконтроллер подобен ПК [25], а из этого следует, что, как и ПК, AVR также может выполнять какую-либо программу, хотя и всего одну в определенный момент времени.

Программа микроконтроллера может храниться во внутренней памяти контроллера и представляет собой серию элементарных команд [26], которые выбирают данные и осуществляют с ними некие операции. В некоторых случаях это означает считывание входящих данных, проверка их состояния и вывода соответствующих выходных значения. Иногда может потребоваться изменение данных и совершение с ними операций, а также передача данных какому-либо внешнему периферийному устройству, например, индикатору, или последовательному порту.

Для таких простых задач используются наборы элементарных команд, каждая из которых имеет аналог на более доступном человеческому восприятию языке. Поэтому наиболее распространенным способом написания программ для контроллера является написание их на языке машинных команд.

Преимуществом машинных команд является очень быстрый, компактный и эффективный код, но создание таких программ одновременно требует и обширных знаний работы процессора микроконтроллера, ручного управления памятью и контроля структуры программы. Поэтому зачастую для написания программ используются другие языки высокого уровня, такие, как Basic, C, и Java. В этом случае задачу по контролю структуры программы и управлению памятью берет на себя компилятор который создает прошивки. Кроме этого, часто используемые функции могут быть при этом помещены в специальные библиотеки и извлекаться из них по мере необходимости.

Микроконтроллеры семейства AVR на сегодняшний день широко используются в компьютерах [27], для автоматизации управления электронной техникой, различными электроприборами и механизмами, применяемыми в промышленных, коммерческих, а также бытовых нуждах. Невысокая стоимость, большой ассортимент и широкие возможности микроконтроллеров этой серии способствовали их большой популярности.

## Выбор Микроконтроллера.

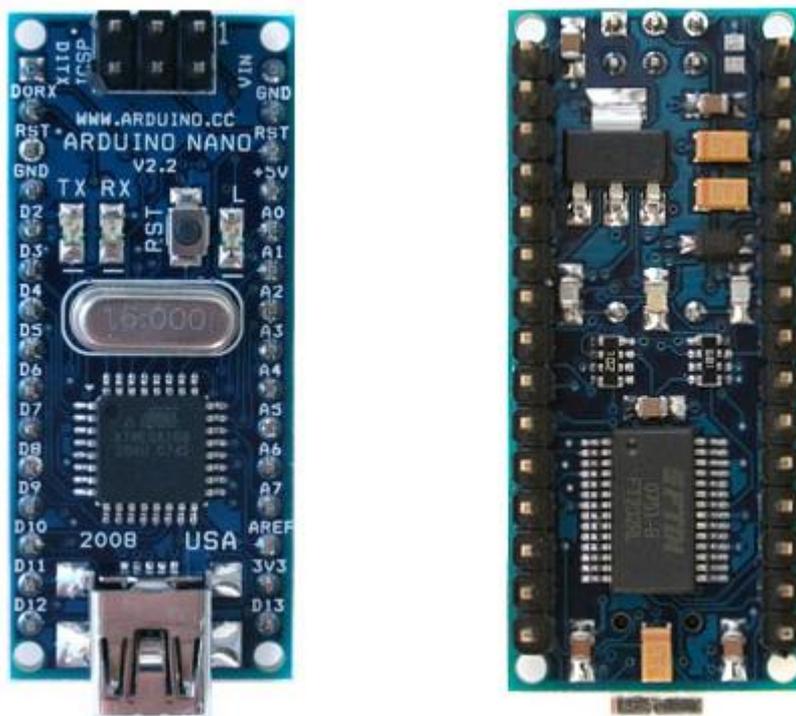


Рисунок 1.6.6 - Ардуино Нано

### Общие сведения

Платформа Nano, построенная на основе микроконтроллере ATmega328 имеет небольшие размеры и может применяться в различных проектах. Она имеет схожую с Arduino Duemilanove структуру, однако отличается внешним видом. Отличие заключается в нехватке силового разъема постоянного тока и работе через кабель MiniUSB.

### Краткие характеристики

Микроконтроллер	Atmel ATmega328
Рабочее напряжение	5 В
Входное напряжение	7-12 В

Входное напряжение	6-20 В
Цифровые Входы/Выходы	14
Аналоговые входы	8
Постоянный ток через вход/выход	30 мА
Флеш-память	32 Кб (2 Кб для загрузчика)
ОЗУ	2 Кб
EEPROM	1 Кб
Тактовая частота	16 МГц
Размеры	1.8 см x 4.1 см

#### Питание:

Arduino Nano получать питание через кабель MiniUSB, или от нерегулируемого 6-20 В (вывод 30), или от стабилизированного 5 В (вывод 27), внешнего источника питания. Автоматически выбирается источник питания с самым высоким напряжением.

Микросхема FTDI FT232RL получает питание, только если сама платформа запитана от USB. Таким образом при работе от внешнего источника, напряжение 3.3 В будет отсутствовать, создаваемое микросхемой FTDI, при этом светодиоды RX и TX мигают только при наличии сигнала на выводах 0 и 1.

## Входы и Выходы

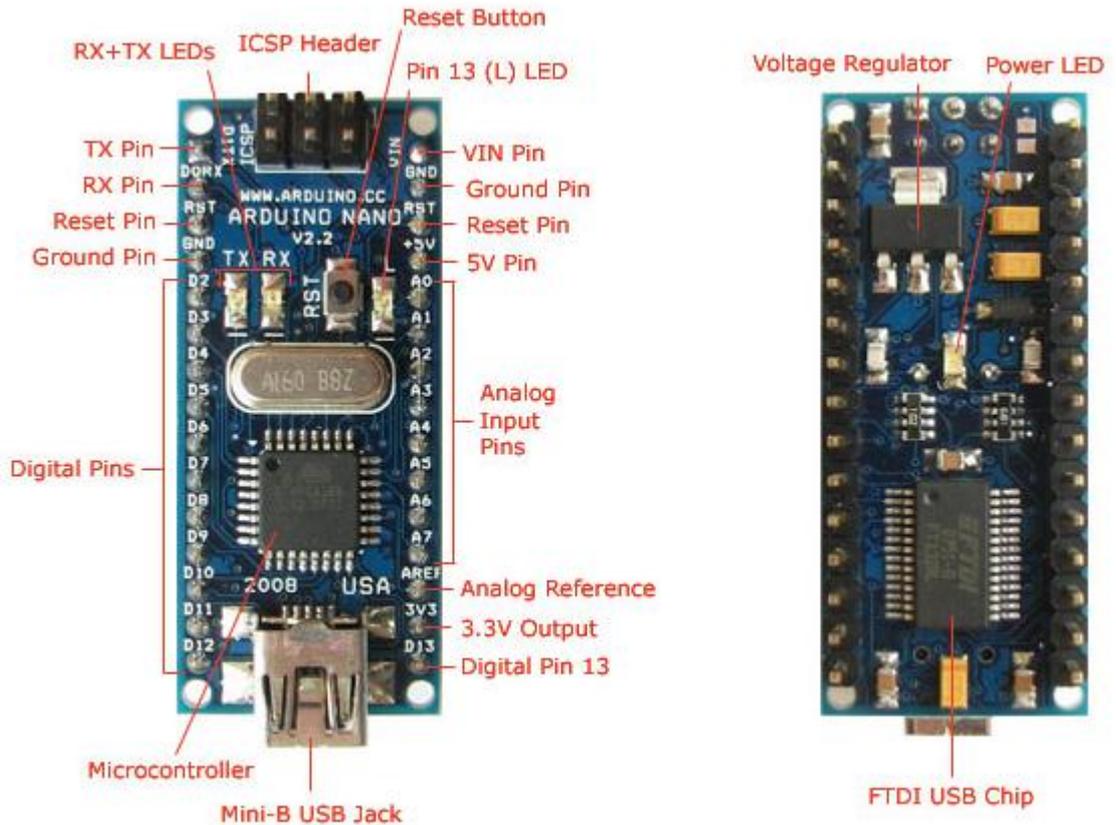


Рисунок 1.6.7 - Выводы микроконтроллера

Каждый из 14 цифровых выходов, используя функции `pinMode()`, `digitalWrite()`, и `digitalRead()`, может настраиваться как вход или выход. Выводы работают только при напряжении 5 В. Каждый вывод имеет резистор (стандартно отключен) 10-40 кОм и может пропускать до 30 мА. Некоторые выводы имеют специфичные функции:

- **Последовательная шина: 0 (RX) и 1 (TX).** Выводы используются для получения и передачи данных.
- **Внешнее прерывание: 2 и 3.** Данные выводы могут быть использованы на вызов прерывания на младшем значении или на переднем, заднем фронте, или при изменении значения.

- **ШИМ: 3, 5, 6, 9, 10, и 11.** Каждый из выводов обеспечивает ШИМ с разрешением до 8 бит при помощи функции `analogWrite()`.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** Посредством данных выводов осуществляется связь SPI.

На платформе Nano установлены 8 аналоговых входов, каждый имеет разрешение 10 бит. Стандартно выводы обладают диапазоном измерения до 5 В относительно нуля, не смотря на это имеется возможность изменить верхнее значение посредством функции `analogReference()`. Некоторые выводы имеют специфичные функции:

- **I2C: A4 (SDA) и A5 (SCL).** Посредством выводов могут осуществляться связь I2C (TWI).

Дополнительная пара выводов платформы:

- **AREF.** Необходимо для аналоговых входов. Используется с функцией `analogReference()`.
- **Reset.** Для перезагрузки микроконтроллер. Обычно применяется для подключения кнопки перезагрузки на платах расширения, закрывающей доступ к кнопке на самой плате.

**Связь**

На платформе Arduino Nano установлено множество устройств для реализации связи с компьютером, другими устройствами Arduino или микроконтроллерами. ATmega328 обладают последовательным интерфейсом UART TTL (5 В), осуществляемый выводами RX и TX. Установленная на плате микросхема направляет данный интерфейс через USB, а драйверы предоставляют виртуальный COM порт программе на компьютере. Монитор последовательной шины (Serial Monitor) программы Arduino позволяет отправлять и принимать текстовые данные при подключении. Светодиоды RX и TX на платформе будут указывать обмен информации по TX и RX.

ATmega328 поддерживают интерфейсы I2C (TWI) и SPI. В Arduino имеется библиотека `Wire` для удобства применения шины I2C.

## Программирование

Платформа программируется посредством ПО Arduino 1.6.5. В меню Tools > Board выбираем «Arduino Diecimila, Duemilanove, Nano Микроконтроллер ATmega328 поставляются с записанным загрузчиком, облегчающим запись программ без использования внешних программаторов. Связь осуществляется по протоколу STK500.

Имеется возможность не использовать загрузчик и запрограммировать микроконтроллер через внутрисхемному протоколу выводами блока ICSP

### Автоматическая (программная) перезагрузка

Nano спроектирован таким образом, что перед записью нового программного кода перезагрузка осуществлялась самой программой, а не нажатием кнопки на плате. Одна из ножек FT232RL, осуществляемая управлением потоком данных (DTR), подключена к выводу перезагрузки ATmega328 через конденсатор 100 нФ. Активация данной линии, перезагружает микроконтроллер. Программа Arduino, используя данный функционал, загружает код одним нажатием кнопки Upload в самой среде разработки. Подача сигнала низкого уровня по линии DTR скоординирована с началом записи программы, что сокращает время загрузчика.

Функция применяет еще одно назначение. Перезагрузка Nano происходит каждый раз при подключении к программе Arduino на компьютере. Следующие несколько секунд после перезагрузки работает загрузчик. Во время программирования происходит таймаут нескольких байтов кода во избежание получения платформой некорректных данных [28]. Если производится разовая отладка прошивки, записанного в платформу, или ввод каких-либо других данных при каждом запуске, необходимо убедиться, что программа на компьютере ждет в течение первой секунды перед передачей данных.

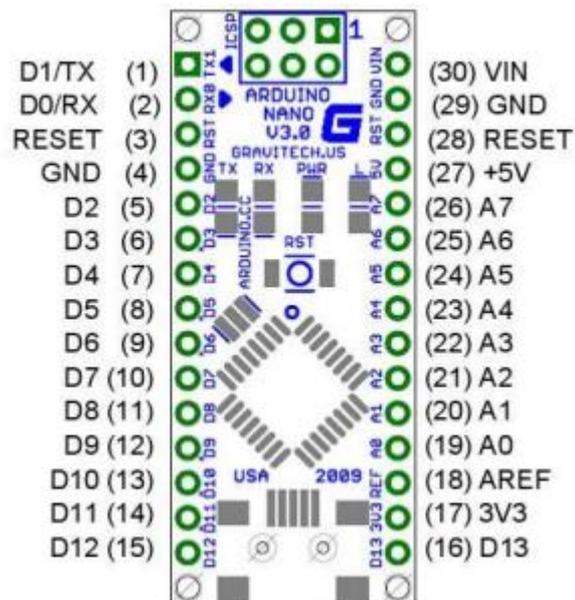


Рисунок 1.6.8 - Обозначение выводов МК

№ вывода	Наименование	Тип	Описание
1 - 2, 5 - 16	D0 - D13	Ввод - вывод	Цифровой ввод-вывод портов D0 - D13
3, 28	RESET	Ввод	Сброс (активный уровень - низкий)
4, 29	GND	Питание	Общий питания
17	3V3	Вывод	+3,3 В с микросхемы FT232
18	AREF	Ввод	Опорное напряжение АЦП
19 - 26	A0 - A7	Ввод	Аналоговый вход, каналы 0 - 7
27	+5V	Ввод - вывод	+5 В на вывод от регулятора на плате, или +5 В на вход от внешнего источника питания

Таблица 1.6.2 - Маркировка выводов микроконтроллеров

Изучив данный материал, мы пришли к выводу, что наш станок может управляться как с помощью персонального компьютера [29], так и с помощью автономного пульта управления выполненного на основе универсальной платформы Arduino Nano.

## 2 Расчетный раздел

### 2.1 Расчет ограничения выходного тока драйвера

Для выполнения постоянного тока привода , опорный ток должен быть отрегулирован с помощью внешнего резистора . Зарядка прекращается когда напряжение NFA ( NFB ) достигает 0,5 В ( установка крутящего момента составляет 100% ), так что ток не превышает заранее определенный уровень .

Для того, чтобы установить пиковый ток до 3.5 А , величина внешнего резистора должна быть рассчитана по формуле:

$$I_{OUT} (A) = 0,5 (V) / R_{NF} (\Omega)$$

, где  $I_{OUT}$  - выходной ток ,  $R_{NF}$  - резистор в цепи стока драйвера

$$R_{NF} = 0,5 / 3,5 = 0,14 \text{ Ом}$$

### **3 Конструкторско-технологический раздел.**

Составили принципиальную схему станка с числовым программным управлением [30]. Соответственно схеме принципиальной сделали разводку печатной платы для разрабатываемого устройства.

В станке применяется двухсторонняя печатная плата выполненная на одностороннем стеклотекстолите. Токпроводящие слои платы делаются комбинированным методом. Проводники изготавливаются методом травления фольгированной подложки, а отверстия с металлизацией - методом электрохимии. Вывод элемента и крепление проводников на печатной плате делают с помощью контактной площадки выполненной в утолщенном участке с большей шириной.

Размеры станка, его конфигурация и фиксирование печатной платы делаются исходя из его расчетных размеров, пайки, элементной базы, контролируемых данных и технических и экономических показателей.

Внешние размеры платы печатной принимаются с выбором класса точности, обязательно учитывается то, что с укрупнением размеров, будут расти погрешности, допуск при исполнении конструктивных элементов для фиксации их на печатной плате. Изготавливаемая печатная плата имеет форму в виде простого прямоугольника.

Станок представляет собой конструкцию с печатной платой размерами 200 x 150 мм с размещенными на ней электронными деталями и контактами для присоединения источника питания и силовых цепей.

Для конструирования электрохимическим методом печатных плат берется листовая материал с изоляционным основанием и нанесенной на него металлической фольгой с двух сторон. В данной магистерской работе выбираем стеклотекстолит фольгированный марки СФ-2-35. Нужными

параметрами номинальных толщин двухсторонних печатных плат являются: 0,5; 1; 1,5; 2; 2,5; 3 (мм). Берем значения  $H_T$  равное 2 мм.

Монтажное отверстие находится в зоне контактной площадки, в которое будет вставляться вывод элемента схемы. Монтажное отверстие имеет металлизированные боковые стенки. Применение отверстий с металлизацией необходимо для большей надежности печатной платы.

Диаметр самих отверстий сделанных в печатной плате должен быть больше диаметра, вставляемого в него вывода элемента, это дает возможность беспрепятственного монтажа устанавливаемого элемента на плату. Размер отверстия с металлизацией влияет от толщины используемой платы. Это связано с тем, что при осаждении металла с помощью гальваники на стенках малого размера, выполненного с толстым основанием, неравномерной получается толщина слоя металлизации, непокрытыми могут остаться места с большим соотношением длины к диаметру отверстий. Размер отверстия с металлизацией необходимо выбирать с условием что он будет составлять половину от толщины платы.

Размер допустимой минимальной ширины проводника на печатной плате выбираем равное 1мм, минимальное расстояние в пределах допуска между элементами выбираем равное 1,5мм.

Проводники на печатной плате изготавливаются с одинаковой шириной во всем диапазоне разводки. Сильноточные цепи силовых элементов, и управления, выполняются по другому принципу [31], исходя из нагрузки на их токопроводящие цепи.

Печатные проводники размещаются равномерным монтажом по всей поверхности печатной платы, учитываются следующие требования: координатная сетка должна быть параллельной или под углом, кратным  $15^\circ$ ; направление движения волны припоя должно быть параллельно углу к нему не должен превышать более  $30^\circ$  со стороны пайки элементов, при условии

что проводящий рисунок не покрыт защитной маской со стороны пайки; к контуру проводящей площади он должен находиться перпендикулярно касательной.

Расположение электрокомпонентов на печатной плате выбирается согласно печатному узлу и его конструктивным требованиям. При размещении электрокомпонентов учитываем: их взаимное расположение, рациональную компоновку, дающую более логичную трассировку и убирающее паразитное влияние на слаботочные элементы управления; обеспечить основные технологические требования, предъявляемые к станкам (автоматизированная сборка и пайка, контроль узлов); данными мерами обеспечить повышенную надежность, выполнение высоких массогабаритных показателей, обеспечить высокое быстродействие, выбрать оптимальный температурный режим с применением теплоотвода, предусмотреть возможность хорошей ремонтпригодности.

Края печатной платы изготавливаются параллельно в соответствии с линиями координатной сетки устройства.

Отверстия и электрокомпоненты токопроводящего медного слоя размещаются на печатной плате в соответствии с базой координат сетки. Электрокомпоненты токопроводящего медного слоя размещаются с краю печатной платы на расстоянии больше чем толщина самой платы с учетом допусков и погрешностей при проектировании и учетом линейных размеров печатной платы.

Координатная сетка наносится с шагом 2,54 мм непосредственно на чертеж. Центры отверстий для монтажа располагаются в компонентах сетки координат. При установке на печатную плату электрокомпонента имеющего два вывода и более, при условии что отверстия между его выводами кратно координатной сетке устройства, элемент размещаем выводами в отверстия на узлах сетки устройства. Размещаемый электрокомпонент который не имеет

выводов, размер между которыми кратный шагу координатной сетки устройства, то в узле координатной сетки располагается один вывод, а центр отверстия на горизонтальной и вертикальной линиях координатной сетки под другой вывод .

Изготовление печатных плат - это очень сложный технологический процесс. Рассчитанная конструкция печатной платы устройства и ее чертежи отправляются в лабораторию, там изготавливаются фотообразцы на твердом носителе.

В лаборатории размещены устройства, которые позволяют масштабировать размер платы в несколько раз.

Основания плат в последствии перевозят в цех специальной обработки, там они доводятся до необходимых размеров в соответствии с техзаданием. Выполняется сверление отверстий и разные операции с материалами, электрокомпонентами и основаниями печатных плат.

Как только печатная плата прошла техническую обработку основания платы попадает в серию на производстве, автоматизированными линиями происходит металлизация отверстий, на отверстия осаживается медь. Затем плата лудится и покрывается свинцом и оловом, в последствии происходит травление. В закрытых емкостях с печатной платы снимается поверхностный слой, на печатной плате остается лишь металл - токопроводящие дорожки.

Затем эта печатная плата с токопроводящими дорожками отправляется на отделение для прессовки, там выполняются работы на прессе согласно установочным размерам печатной платы.

Предварительно обработанные печатные платы отправляются на участок оксидирования, в последствии поступают на участок фрезерования [27], где происходит их обработка по контуру в соответствии с расчетными размерами.

После полной обработки основания печатной платы происходит установка электрокомпонентов. Электрокомпоненты на плате размещаются как вручную, или с применением "лазерного щупа": основание печатной платы размещается в специальном пазу монтажного стола, лазерным лучом отмечаются места на печатной плате, где будут размещаться электрокомпоненты одного и того же типа.

Пайка электрокомпонентов будет происходить волной припоя, на участок с жидкой расплавленной смесью свинца и олова погружается печатная плата с установленными электрокомпонентами и при определенной температуре, регулируемой специальными автоматами [32], в отверстиях печатной платы выполняется пайка выводов электрокомпонентов.

Для защиты печатной платы от внешних вредных воздействий окружающей среды и других агрессивных сред, наносится защитная маска. Печатная плата маркируется и передается на контроль в технический отдел ОТК.

## 4 Эффективность проекта

### 4.1 Расчет затрат на основные материалы

Полную стоимость комплектующих изделий определили по таблице 4.1

Наименование изделия	Марка, размер	ГОСТ, ТУ	Количество, шт	Цена за единицу, (руб.)	Затраты, (руб.)
1	2	3	4	5	6
Микросхемы	ТВ6560АНQ	ГОСТ17467-79	3	410	1230
	7805		1	15	15
	7812		1	15	15
	K561ТЛ2		1	15	15
Диоды, стабилитроны, светодиоды, оптопары	РС817	ТУ 362.029	9	12	108
	30ВQ100		12	30	360
	МВR1660		4	26	104
Конденсаторы	ЕСАР 470мкФ 50В	ОЖО.464. 214ТУ	3	16	48
	К73-17 100В 0,1мкФ		27	20	540
	ЕСАР 2200мкФ		3	60	180

	50В				
--	-----	--	--	--	--

Продолжение таблицы 4.1

Наименование Изделия	Марка, размер	ГОСТ, ТУ	Количество, шт	Цена за единицу,(ру б.)	Затрат ы, (руб.)
1	2	3	4	5	6
Резисторы	CRL-5W 0,15Ом ±5%	ГОСТ 7113-77	6	8	48
	МЛТ-0.125- 150Ом±10%		9	2	18
	МЛТ-0.125- 1кОм±10%		3	2	6
	МЛТ-0.125- 4,7кОм±10 %		5	2	10
	МЛТ-0.125- 10кОм±10%		21	2	42
	МЛТ-0.125- 33кОм±10%		18	2	36
	МЛТ-0.125- 6,8МОм±10 %		21	2	42
Клеммник винтовой	5ЕНDV-04P		3	13	39
Клеммник силовой	5ЕСDV-04P		3	35	105
Стеклотекстол ИТ	СФ-2-35	ГОСТ 1 0316-78	1	120	120

--	--	--	--	--	--

Продолжение таблицы 4.1

Наименование Изделия	Марка, размер	ГОСТ, ТУ	Количество, шт	Цена за единицу, (руб.)	Затраты, (руб.)
1	2	3	4	5	6
Разъем	DB25		1	20	20
Алюминиевая станина с направляющими	AL3040		1	35000	35000
Двигатель	NEMA23 HS7430		3	2930	8790
Блок питания шпинделя	DC48		1	1700	1700
Блок питания драйвера	DC30V		1	1600	1600
Гибкая шина	15BF02.30 R28		1	909	909
Кабель экранированный	КГВЭВ 4x1.5		10	117	1170
Концевой выключатель	MSW13+		2	30	60
Концевой выключатель	SC799		4	40	160

Разъем	KP2G16K5P		5	173	865
--------	-----------	--	---	-----	-----

Продолжение таблицы 4.1

Наименование Изделия	Марка, размер	ГОСТ, ТУ	Количество, шт	Цена за единицу, (руб.)	Затраты, (руб.)
1	2	3	4	5	6
Кнопка стоп	XB2- BS542		1	350	350
Шнур сетевой	S-18		1	65	65
Корпус	АТХ-24-1		1	600	600
Держатель светодиодов	LC-05		3	10	30
Шнур сетевой	S-18		1	61	61
Припой	Sn60/ Pb40		1	50	50
Флюс	ЛТИ-120		1	30	30
Провод монтажный	FRC1-10- 30		3	13	39
Провод монтажный силовой	ПУВ1-4		2	14	28
Шпиндель	HXKJGS52		1	4000	4000

Итого					58608
-------	--	--	--	--	-------

Стоимость покупных комплектующих изделий с учетом транспортно-заготовительных расходов:

$$C_{\text{покуп}} = \sum_{i=1}^n S_{\text{покуп}} \cdot (1 + K_{\text{мз}}),$$

где  $K_{\text{ТЗ}}$  – коэффициент транспортно-заготовительных расходов;  $K_{\text{ТЗ}} = 0,04$

$S_{\text{покуп}}$  – стоимость покупных комплектующих изделий, руб.

$$C_{\text{покуп}} = 58608 \times (1 + 0,04) = 60952,32 \text{ руб.}$$

Цены актуальны на 22.05.2016г. Валютный курс составлял 65 рублей 41 копейка за 1 доллар. Закупка проводилась в магазинах:

<http://www.chipidip.ru> <http://www.ebay.com>

## **Заключение**

В результате проведенных исследований и расчетов было спроектирован станок с числовым программным управлением. Согласно техническому заданию данное устройство способно может выполнять обработку различных материалов, мягких металлов. Это подтверждено практической постройкой системы. В ходе проектирования был использован ШИМ преобразователь, что позволило эффективно использовать источник питания ограниченной мощности, понизить паразитный резонансный эффект путем изменения рекомендуемой схемы включения заводом изготовителем.

## Литература

- 1 Семенкова О.И. Автоматизация проектно-конструкторских работ и технологической подготовки производства в машиностроении. Т. 1- Минск: Высшая школа, 2005
- 2 Фролова К.В., Воробьева Е.И.. Механика промышленных роботов: Учеб. Пособие для вузов: В 3 кн. / Под ред. Кн.2: расчет и проектирование механизмов – М: Высш. шк., 1988
- 3 Гольдштейн А.И., Молочник В.И. О внутренней структуре постпроцессоров. — В кн.: Повышение эффективности использования станков с ЧПУ. - Киев: Знание, 2006.
- 4 Лещенко А.И. , Программирование и технологические процессы для станков с ЧПУ, Мариуполь, ПГТУ, 2005
- 5 Петров И.В. , Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования , СОЛОН-Пресс , 2004
- 6  
Ловыгин А. А., Теверовский Л. В. Современный станок с ЧПУ и CAD & CAM система - 2012
- 7 Гурьянихин В.Ф. , Проектирование технологических процессов обработки заготовок на станках с ЧПУ, Ульяновск, УлГТУ, 2007
- 8 Каштальян И.А. ,Обработка на станках с числовым программным управлением ,Высшэйшая школа ,1989
- 9 Михайлов О.П. ,Современный электропривод станков с ЧПУ и промышленных роботов , Высшая школа,1989

10 Сотников В.И. , Программирование и работа на станках, оснащенных системой ЧПУ, ОрелГТУ , 2009

11 Фадюшин И.Л. , Инструмент для станков с ЧПУ, многоцелевых станков и ГПС, Машиностроение ,1990

12 Кузнецов Ю.И., Оснастка для станков с ЧПУ, Справочник, Машиностроение , 1990

13 Комиссаров В.И.,Фильченко Ю.А., Юшкевич В.В. Размерная наладка станков с ЧПУ на роботизированных участках, Владивосток , ДВПИ, 1985.

14 Аверьянов О.И. Модульный принцип построения станков с ЧПУ, Машиностроение ,1987

15 Гжиров Р.И., Серебренницкий П.П. Программирование обработки на станках с ЧПУ. Справочник, Л. Машиностроение, 1990

16 Кошкин В.Л. , Аппаратные системы числового программного управления , Машиностроение , 1989

17 Общемашиностроительные нормативы времени и режимов резания для нормирования работ, выполняемых на универсальных и многоцелевых станках с ЧПУ. Часть2, М.,Экономика, 1990

18 Головенков С.Н. , Основы автоматизации и автоматического регулирования станков с программным управлением, Машиностроение 1980

19 Общемашиностроительные нормативы времени и режимов резания для нормирования работ, выполняемых на универсальных и многоцелевых станках с ЧПУ. В 2-х ч., Экономика, 1990 20

Жигало Н.И., Яцура Е.С. Обработка материалов, станки и инструменты. 1984

21 Фельдштейн Е.Э., Обработка деталей на станках с ЧПУ, Новое знание, 2008

22 Байков В.Д., Решение траекторных задач в микропроцессорных системах ЧПУ, Машиностроение, 1986

23 Монахов Г.А., Станки с программным управлением. Справочник, Машиностроение, 1975

24 Кенио Т. Шаговые двигатели и их микропроцессорные системы управления, 1987

25 Кошкин В.Л. Аппаратные системы числового программного управления; 1989

26 Лещенко В.А., Станки с числовым программным управлением (специализированные), Машиностроение, 1988

27 Должиков В.П., Основы программирования и наладки станков с ЧПУ, Томск, ТПУ, 2011

28 Дж. Вильямс Программируемые роботы - М.: NT Press, 2006

29 Павлов С. Механика самодельного станка ЧПУ, PureLogic RND Russia, 2008

30 Андреев Г.И. Работа на токарных станках с ЧПУ, Ирлен Инжиниринг, 2005

31 Кряжев Д.Ю. Фрезерная обработка на станках с ЧПУ, Ирлен Инжиниринг, 2005

32    Филенко Н. Станок с ЧПУ своими руками, PureLogic RND Russia,  
2008