Titanium®

Часть I

Руководство по технологическому программированию

Оглавление

Введение

О настоящем руководстве

Данное руководство предназначено для операторов станков и для наладчиков, занимающихся настройкой системы УЧПУ Titanium®. В руководстве приводится справочная информация по программированию управляющих программ, по которым будет происходить обработка заготовок. Также приводится информация по программированию макроязыка, являющегося расширением стандартного языка G-кодов и реализующего его недостающие функциональные возможности.

Специальные обозначение

Записи формата **X__, Y__, Z__** и т. д. обозначают указание названий аргументов и их значений. Например, **X** — название аргумента G-кода, а вместо пропуска («__») можно подставлять численные значения (например, X10.5).

Запись **N1...N1000** обозначает диапазон обозначений: N1, N2, N3, N4, N5, ..., N998, N999, N1000. В качестве другого примера: запись **G53**...**G59** будет соответствовать диапазону обозначений: G53, G54, G55, G56, G57, G58, G59.

Запись вида **N5n12** обозначает любой параметр из набора: N5012, N5112, N5212, N5312, N5412, N5512, N5612, N5712, N5812, N5912. То есть вместо буквы «n» в названии параметра подставляется цифра.

Под декартовой осью понимается ось станка, в параметре N5n22 которой выставлено, вдоль какой оси ось станка расположена.

1 Общие сведения

1.1 Работа за станком

Для обработки детали на станке с УЧПУ необходимо создать управляющую программу, а затем приступить к работе на станке с использованием данной программы. Процесс создания программы описан в разделе 2 «Программирование управляющих программ». На рисунке 1.1.1 показана общая последовательность действия для начала работы на станке.

Рис. 1.1.1: Обработка детали на станке

Перед созданием управляющей программы необходимо составить план обработки, который должен включать в себя следующие пункты.

1. Размеры заготовки и её положение.
2. Метод закрепления заготовки.
3. Разбиение процесса обработки на отдельные этапы.
4. Используемые при обработке инструменты.
5. Обработка.

Для каждого этапа необходимо выбрать метод обработки (черновой, получистовой, чистовой) и подобрать соответствующую рабочую подачу.

1.2 Общие сведения о УЧПУ

При работе с УЧПУ оператор использует программы, сохраняемые на постоянные носители информации. Однако все переменные (кроме специально отведённых), используемые в программе, не сохраняются в энергонезависимой памяти и будут утеряны при перезапуске системы УЧПУ. Также при резком отключении питания системы УЧПУ значения переменных, сохраняемых в энергонезависимой памяти, могут иметь значения, отличные от последних заданных (вследствие особенностей работы операционной системы). Поэтому не рекомендуется отключать питание системы УЧПУ во время исполнения управляющей программы. При резком отключении питания системы УЧПУ возможна потеря информации, с которой работала УЧПУ в момент отключения. Поэтому рекомендуется время от времени создавать резервные копии важных данных (управляющих программ, привязок деталей и т. п.).

Все файлы, с которыми работает оператор, сохраняются в его домашнем каталоге. Для переноса настроек УЧПУ на другой станок достаточно скопировать все файлы (кроме каталога конфигурации) из домашнего каталога данной УЧПУ в другую систему УЧПУ. Таким же образом можно сделать резервную копию настроек станка. Для этого необходимо сохранить все файлы из домашнего каталога на внешний носитель информации (flash-накопитель).

После создания и отладки управляющих программ по обработке деталей можно выставить на них права доступа только для чтения, чтобы предотвратить случайное изменение программ оператором станка. В дальнейшей возможно снова вернуть право доступа на запись в случае необходимости.

2 Программирование управляющих программ

2.1 Управляемые оси

2.1.1. Наименование оси

Имя оси прописывается в параметре N5n00 и может содержать от 1 до 10 символов латинского алфавита. Рекомендуется названия осей обозначать одной, либо двумя буквами. Основные 3 оси принято обозначать буквами X, Y и Z. Дополнительные оси можно обозначать буквами A, B, C, U, V или W. Буквы A и B подходят для обозначения круговых осей, буквы U, V и W — для обозначения осей 2-го суппорта. Буква C рекомендуется для именования шпинделя, работающего в режиме оси.

Каждой оси можно присвоить номер в декартовой системе координат (параметр N5n22). Соответствие номеров и декартовых осей приведено в таблице 2.1.1.

Примечание. Имена осей не могут повторяться. В случае указания одинаковых названий осей в параметрах будет работать только первая из указанных осей.

Таблица 2.1.1: Соответствие декартовым осям

Номер	Декартова ось
0	–
1	X
2	Y
3	Z

2.1.2. Единицы измерения перемещений

Для каждой оси с помощью делителя (параметр N5n13) и умножителя (параметр N5n14), указанных в параметрах оси, формируется минимальная величина программируемого перемещения.

По умолчанию используется метрическая система координат. В параметрах необходимо задать соответствие между минимальной величиной перемещения и величиной перемещения в 1 мкм.

Для корректного перемещения оси в режиме управления по скорости необходимо также правильно выставить добротность привода (параметр N6n08). То есть расстояние, которое проедет привод без включённого ПИД-регулятора, должно наиболее точно соответствовать расстоянию, полученному с энкодера двигателя. Для цифровых приводов идеальное значение добротности часто соответствует числу ± 16.66666666666667 ($\pm 16.(6)$).

2.2 Программные каналы и суппорта

В системе ЧПУ предусмотрена возможность параллельной работы двух и более каналов. Для каждого канала в параметрах можно задавать оси, доступные для интерполирования. Оси в разных каналах интерполируются независимо друг от друга. Синхронизацию между суппортами можно производить через М-команды (при наличии таковых в модуле ПЛК). Также для каждого канала задаются свои плоскости интерполирования (G17, G18, G19).

В рамках канала можно выбирать активную **группу осей** данного канала. Группа осей позволяет исключить из канала определённые оси (например, если они используются модулем ПЛК). Также в рамках группы осей можно переопределить стандартные плоскости интерполирования (G17, G18, G19). Если ни одна группа осей не выбрана, то используются параметры канала по умолчанию:

- разрешены все доступные для канала оси;
- используются плоскости интерполирования канала по умолчанию.

Помимо каналов существует такое понятие как суппорт. В каждом канале одновременно могут быть активными только два суппорта:

- суппорт, на котором установлена обрабатываемая деталь;
- суппорт, на котором находится обрабатывающий инструмент.

В рамках канала предусмотрена возможность выбирать активные суппорта, если у станка более двух суппортов. Таким образом, на токарном станке можно чередовать обработку детали фрезой и резцом, если они закреплены на разных суппортах.

2.3 Описание G-кодов

2.3.1. G-функция

G-функции определяют команды и флаги, изменяющие поведение команд. G-функция состоит из буквы G и номера, следующего за ней. Номер определяет код G-функции. Коды бывают модальные и однократные.

Модальные G-коды сохраняют своё значение до конца работа программы, пока не будут отменены другим G-кодом либо командой макроязыка.

Однократные G-коды исполняются один раз[1] в рамках текущего кадра.

G-коды делятся на группы. Каждая группа включает в себя объединённые по функциональному назначению G-коды, из которых одновременно может быть выбран только один.

Пример. Модальные G-коды

```
1 G1 X10 Y10 F100 ;выполняем команду G1, выставляется в качестве текущего
    флаг G1
2 X5 ;выполняется G1
3 Y0 ;выполняется G1
4 G0 X0 ;выполняем команду G0, выставляется в качестве текущего флаг G0
5 X10 Y10 ;выполняется G0
```

Пример. Однократные G-коды

```
1 G1 ;задали текущим флагом G1
2 G28 X0 Y0 ;переходим в координаты нуля станка, однократный G-код
3 X10 Y10 ;выполняется G1
```

2.3.2. Перечень G-кодов

Все G-коды делятся на системные группы по функциональному назначению. Единовременно может быть активен только один флаг из группы.

Каждый из G-кодов имеет определённое назначение. В таблице 2.3.1 приведено описание основных G-кодов и их распределение по группам.

[1] Многопроходные циклы и подпрограммы являются однократными G-кодами, но охватывают много кадров.

Таблица 2.3.1: Перечень G-кодов

G-код	Группа	Назначение
1	2	3
G02	01	Круговая интерполяция по часовой стрелке
G00	01	Позиционирование
G01	01	Линейная интерполяция
G03	01	Круговая интерполяция против часовой стрелки
G04	00	Задержка по времени
G09	00	Точный останов в конце кадра
G17	02	Выбор декартовой плоскости XY (1, 2)
G18	02	Выбор декартовой плоскости ZX (3, 2)
G19	02	Выбор декартовой плоскости YZ (2, 3)
G27	00	Возврат в первый ноль станка
G28	00	Возврат в основное положение станка
G28.1	00	Выход в ноль по аппаратным выключателям
G29	00	Возврат в из референтной позиции
G30	00	Возврат во 2-й ноль станка
G31	00	Функция пропуска
G33	01	Нарезание резьбы
G40	07	Отмена коррекции на режущий инструмент
G41	07	Коррекция на режущий инструмент слева
G42	07	Коррекция на режущий инструмент справа
G43	08	Коррекция на длину инструмента в «+»
G44	08	Коррекция на длину инструмента в «-»
G49	08	Отмена коррекции на длину инструмента
G50	22	Отмена масштабирования системы координат
G51	22	Масштабирование системы координат

G-код	Группа	Назначение
1	2	3
G52	28	Смещение текущей системы координат
G53	27	Перемещение в машинные координаты
G54	14	Выбор системы координат заготовки 1
G55	14	Выбор системы координат заготовки 2
G56	14	Выбор системы координат заготовки 3
G57	14	Выбор системы координат заготовки 4
G58	14	Выбор системы координат заготовки 5
G59	14	Выбор системы координат заготовки 6
G61	15	Режим точного останова в конце кадра
G62	15	Режим автоматической стыковки кадров по скорости
G63	15	Режим нарезания резьбы
G64	15	Режим грубой обработки
G65	00	Вызов макропрограммы
G66	12	Вызов модальной макропрограммы
G67	12	Отмена модальной пользовательской макропрограммы
G68	16	Поворот системы координат в текущей плоскости
G68.2	29	Поворот системы координат в пространстве
G70	09	Многопроходной цикл чистовой обработки по профилю
G71	09	Многопроходной цикл продольного съёма по профилю
G72	09	Многопроходной цикл поперечного съёма по профилю
G74	09	Многопроходной цикл горизонтальной расточки
G75	09	Многопроходной цикл вертикальной расточки
G76	09	Многопроходной цикл нарезания резьбы
G80	09	Отмена модальной макропрограммы
G81	09	Многопроходный цикл сверления с дроблением стружки
G82	09	Многопроходный цикл сверления с паузой

G-код	Группа	Назначение
1	2	3
G83	09	Многопроходный цикл сверления с выводом сверла
G84	09	Цикл нарезания резьбы метчиком
G85	09	Фиксированный цикл растачивания
G90	03	Режим рабочих координат
G91	03	Режим приращений координат
G90.1	25	Режим рабочих координат для параметров I, J, K
G91.1	25	Режим приращений координат для параметров I, J, K
G92	18	Задание пользовательской системы координат
G94	05	Минутная подача
G95	05	Оборотная подача
G96	13	Контроль постоянства скорости резания
G97	13	Отмена контроля постоянства скорости резания
G101	09	Перемещение по вектору
G103		Стандартные фигуры
G111		Фаска по длине и катету
G112		Галтель (по G2) между точками
G176		Цикл нарезания круглопрофильной резьбы
G181	09	Цикл расточки отверстия фрезой
G197		Заглубление фрезой по замкнутому контуру
G220	02	Выбор плоскости с произвольными осями
G221	19	Разрешение процентовки подачи
G222	19	Запрет процентовки подачи
G223	23	Разрешение процентовки шпинделя
G224	23	Запрет процентовки шпинделя
G301		Прямоугольный карман

2.4 Интерполяция

2.4.1. Позиционирование (G0)

Команда G0 перемещает инструмент в заданное положение. Положение можно задавать в абсолютных величинах либо через приращения относительно текущих координат.

Формат команды:

```
G0 X__ Y__ ...
```

В качестве аргументов указываются названия осей и координаты, в которые необходимо переместиться.

Пример. Команда G0

```
G90 ;абсолютная система координат
G0 X10 Y10 Z10 ;перемещение в координаты X=10, Y=10, Z=10
X0 Y0 ;перемещение в координаты X=0, Y=0, Z=10
Z0 ;перемещение в координаты X=0, Y=0, Z=0
```

В соответствии с параметром N3033 движение может быть интерполируемым либо асинхронным.

Если параметр N3033 = 0, то позиционирование осуществляется асинхронно по осям со скоростью ускоренного подвода. Данный режим используется по умолчанию.

Если параметр N3033 = 1, то позиционирование осуществляется с интерполяцией по осям аналогично команде G1. Скорость, с которой движется инструмент, вычисляется, исходя из значений скоростей ускоренного подвода по каждой из осей, и является максимально возможной. Траектория движения инструмента отличается от траектории движения линейной интерполяции (G1). Наглядно разница продемонстрирована на рисунке 2.4.1.

Скорость форсированной продольной подачи в команде G0 устанавливается изготовителем станка для каждой оси отдельно в параметре N5n12. В режиме позиционирования в начале блока происходит ускорение инструмента до предварительно заданной скорости, а в конце блока — замедление. Программа переходит к выполнению следующего блока после подтверждения выхода в заданную позицию. «Выход в заданную позицию» означает, что двигатель подачи находится в заданном диапазоне. Этот диапазон устанавливается изготовителем станка в параметре No6n10.

Примечание. Скорость ускоренного подвода не может задаваться в аргументе F. Даже если задается позиционирование при линейной интерполяции, аргумент F на неё не влияет. Обязательно убедитесь в том, что инструмент не ударится о заготовку при позиционировании по G28, G30 или G0.

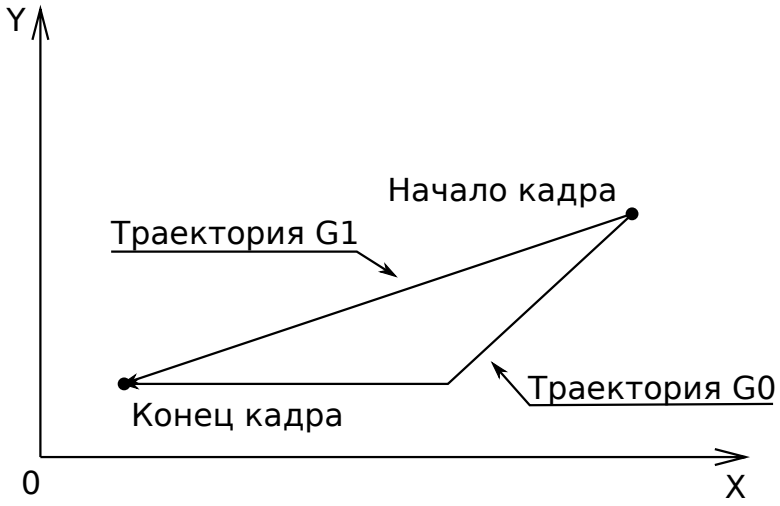


Рис. 2.4.1: Отличия траекторий G0 и G1

2.4.2. Линейная интерполяция (G1)

При линейной интерполяции инструмент перемещается вдоль прямой линии[2], координаты которой задаются аргументами функции G1.
Формат команды:

```
G1 X__ Y__ ... F__
```

Для использования линейной интерполяции необходимо указать подачу с помощью аргумента F. Подача определяет скорость, с которой будет двигаться инструмент по прямой линии.

Инструмент перемещается вдоль линии в заданное положение со скоростью подачи, заданной в F. Скорость подачи, заданная в F, действует до ввода нового значения. Нет необходимости задавать ее в каждом блоке. Скорость подачи, заданная F-кодом, задаётся вдоль траектории движения инструмента. Если F-код не задан, скорость подачи считается равной нулю. Скорость подачи в направлении каждой оси вычисляется по формулам, приведённым ниже.

$$L = \sqrt[2]{\sum_{i=1}^{N} \delta_i^2} \tag{2.4.1}$$

$$F_i = \frac{\delta_i}{L} \cdot F \tag{2.4.2}$$

Где F — заданная подача; i — порядковый номер оси; δ_i — величина, на которую будет произведено перемещение по i-той оси; N — количество интерполируемых осей; F_i — скорость, с которой будет производиться пере-

[2]Интерполировать можно линейные оси вместе с круговыми, в таком случае траектория движения будет отличаться от прямой линии.

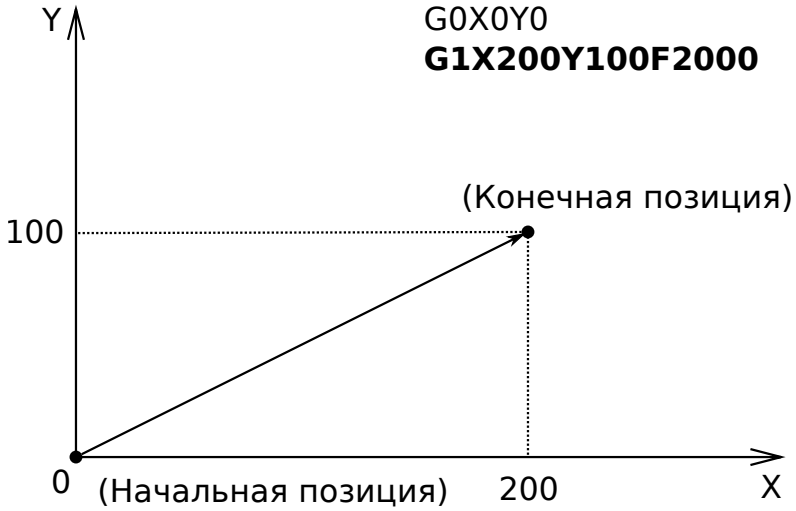


Рис. 2.4.2: Графическая интерпретация кадра G1

мещение по i-й оси.

Пример. Расчёт кадра G91G1X5Y10B20F1000

$$L = \sqrt[2]{\sum_{i=1}^{3} \delta_i^2} = \sqrt[2]{5^2 + 10^2 + 20^2} = 22.913$$

$$F_X = F_1 = \frac{5}{22.913} \cdot 1000 = 218.218$$

$$F_Y = F_2 = \frac{10}{22.913} \cdot 1000 = 436.436$$

$$F_B = F_3 = \frac{20}{22.913} \cdot 1000 = 872.872$$

На рисунке 2.4.2 показан графический пример линейной интерполяции по двум осям.

2.4.3. Круговая/винтовая интерполяция (G2, G3)

Круговая интерполяция осуществляет перемещение инструмента по осям выбранной плоскости по дуге окружности. При этом по остальным осям, если они указаны, осуществляется линейная интерполяция.

Формат команды в плоскости XY:

```
G17 G2 X__ Y__ I__ J__ P__ F__ ;указание координат центра дуги
```

```
G17 G2 X__ Y__ R__ P__ F__ ;указание радиуса дуги
```

Таблица 2.4.1: Описание формата команды

Команда	Описание
G17	Дуга в декартовой плоскости XY
G18	Дуга в декартовой плоскости ZX
G19	Дуга в декартовой плоскости YZ
G02	Движение по часовой стрелке
G03	Движение против часовой стрелки
G90.1	Координаты I, J, K задаются как абсолютные (N3021 = 0)
G91.1	Координаты I, J, K задаются как относительные (N3021 = 0)
I__	Координата центра окружности декартовой оси X
J__	Координата центра окружности декартовой оси Y
K__	Координата центра окружности декартовой оси Z
R__	Радиус дуги окружности
P__	Полное количество оборотов по окружности
F__	Скорость подачи по дуге

Формат команды в плоскости ZX:

```
G18 G2 X__ Z__ I__ K__ P__ F__ ;указание координат центра дуги
```

```
G18 G2 X__ Z__ R__ P__ F__ ;указание радиуса дуги
```

Формат команды в плоскости YZ:

```
G19 G2 Z__ Y__ J__ K__ P__ F__ ;указание координат центра дуги
```

```
G19 G2 Z__ Y__ R__ P__ F__ ;указание радиуса дуги
```

Формат команды в произвольной плоскости:

```
G220 A0 B0 ;работаем в плоскости AB
G2 A__ B__ I__ J__ P__ F__ ;указание координат центра дуги
```

```
G220 A0 B0
G2 A__ B__ R__ P__ F__ ;указание радиуса дуги
```

Для команды G3 формат аналогичный. Описание флагов и аргументов команды приведено в таблице 2.4.1.

Аргумент P задаёт полное количество оборотов, которое должен сделать инструмент по задаваемой окружности. Заданные обороты начинают выполняться после того, как инструмент выйдет в конечную точку дуги. Аргумент P не является обязательным. Если аргумент P не был указан, то его значение приравнивается к нулю.

Подача F может быть задана до команды G2 или G3.

Если одновременно заданы и радиус R, и координаты центра дуги I, J или K, то для расчёта центра берётся радиус, а координаты центра дуги игнорируются.

При параметре N3021, выставленном в 0, модальные команды G90.1 и G91.1 определяют, как будут задаваться координаты центра через I, J, K. Если задана модальная команда G90.1, то аргументы I, J, K задают координаты центра в рабочей системе координат. Если же задан флаг G91.1, то аргументы I, J, K представляют из себя смещения центра координат относительно координат начала дуги (текущей позиции инструмента).

Если параметр N3021 = 1, то на аргументы I, J, K влияют модальные команды G90 и G91.

Если параметр N3021 = 90, то I, J, K всегда задаются в абсолютных координатах.

Если параметр N3021 = 91, то I, J, K всегда задаются в относительных координатах.

Иногда требуется произвести движение по полному обороту окружности. Для этого необходимо выйти в координату, находящуюся на окружности, а затем выполнить круговую интерполяцию, указав в качестве конца дуги те же координаты, что и текущие. Помимо этого необходимо явно указать координаты центра (через I, J, K).

Пример движения по полной окружности

```
1 G90 G0 X10 Y10 Z3
2 G1 Z0
3 G17 G91.1 G2 X10 Y10 I10 ;выточить окружность радиусом 10, центр в
    координатах X=20, Y=10
```

Направление круговой интерполяции

«По часовой стрелке» (G02) и «против часовой стрелки» (G03) в плоскости XY (плоскости ZX или плоскости YZ) определяется, если посмотреть на плоскость XY в направлении от плюса к минусу по оси Z (оси Y или оси X соответственно) в декартовой системе координат. Пояснения приведены на рис. 2.4.3.

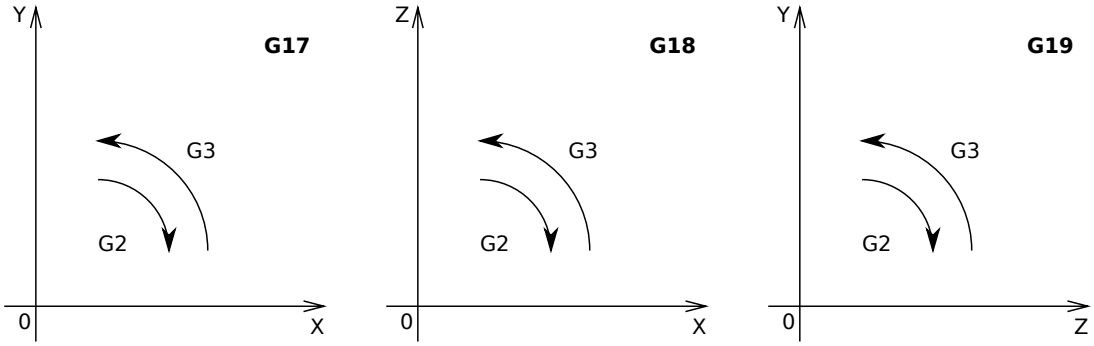


Рис. 2.4.3: Направление движения по декартовым осям

Ограничения

- Если одновременно задаются I, J, K и R, то задается дуга, задаваемая адресом R, а другие дуги пропускаются.
- Если в кадре круговой интерполяции не задано ни одной оси текущей декартовой плоскости, то выдаётся сигнал тревоги и посылается сигнал сброса управляющей программы.
- Если в кадре круговой интерполяции задана только одна ось, принадлежащая текущей декартовой плоскости, то в качестве значения второй берётся её текущая координата.
- Если не были указаны ни радиус (R), ни центр дуги (I, J, K), то выдаётся сигнал тревоги и посылается сигнал сброса управляющей программы.
- При указании радиуса (R), меньшего половины расстояния от текущей точки до заданной, выдаётся сигнал тревоги и посылается сигнал сброса управляющей программы.
- При указании центра дуги (I, J, K), если радиус начала кадра и радиус конца кадра отличаются на значение, большее параметра N1041, то выдаётся сигнал тревоги и посылается сигнал сброса управляющей программы.

Использование

На рисунке 2.4.4 показан пример двух кадров круговой интерполяции в декартовой плоскости XY.

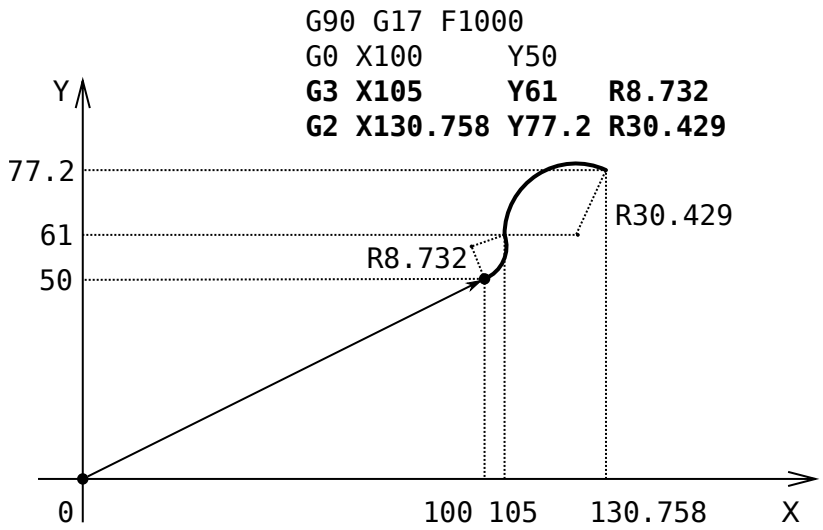


Рис. 2.4.4: Пример круговой интерполяции

Пример. Варианты круговой интерполяции

```
( В абсолютной системе координат )
G0 X200.0 Y40.0 Z0
G90 G91.1 G03 X140.0 Y100.0 R60.0 F300
G02 X120.0 Y60.0 R50.0
( или )
G0 X200.0 Y40.0 Z0
G90 G91.1 G03 X140.0 Y100.0 I-60.0 F300
G02 X120.0 Y60.0 I-50.0

( То же самое в относительной системе координат )
G91 G03 X-60.0 Y60.0 R60.0 F300
G02 G91.1 X-20.0 Y-40.0 R50.0
( или )
G91 G03 X-60.0 Y60.0 I-60.0 F300
G02 G91.1 X-20.0 Y-40.0 I-50.0
```

Круговая интерполяция производится согласно следующим формулам:

$$F_x = F_c \cos \alpha \tag{2.4.3}$$

$$F_y = F_c \sin \alpha \tag{2.4.4}$$

F_c — подача движения по дуге окружности. При круговой интерполяции $F_c = F$;

F_x — подача первой оси плоскости интерполирования;

F_y — подача второй оси плоскости интерполирования;

α — текущий угол поворота вокруг центра дуги.

Винтовая интерполяция

Винтовая интерполяция происходит, если в круговой интерполяции указать координаты осей, не входящих в плоскость интерполирования. В результате по двум осям будет производиться круговая интерполяция, а по остальным, которые были указаны, будет производиться линейная интерполяция.

Подача винтовой интерполяции по каждой из линейный осей определяется по следующей формуле.

$$F_i = \frac{\delta_i}{L} \cdot F \tag{2.4.5}$$

$$L = \sqrt{L_c^2 + L_l^2} \tag{2.4.6}$$

$$L_c = 2\pi R \frac{\triangle\alpha}{360} \tag{2.4.7}$$

$$\triangle\alpha = \alpha_A - \alpha_B + 360P \tag{2.4.8}$$

L — общая длина траектории перемещения;

L_c — длина дуги окружности круговой составляющей перемещения по осям;

L_l — длина траектории линейной составляющей перемещения по осям;

δ_i — величина перемещения по оси;

F_i — подача по линейной оси;

$\triangle\alpha$ — величина угла, на который необходимо будет произвести перемещение от начала до конца кадра;

α_A — угол начала круговой интерполяции;

α_B — угол конца круговой интерполяции;

P — количество полных оборотов (указываются в аргумента P__ круговой интерполяции).

2.4.4. Нарезание резьбы (G33)

Команда G33 предназначена для нарезания резьбы с постоянным шагом. Команда ожидает, пока шпиндель не проскочит через свою 0-метку, и только после этого продолжает работу программы. Если в качестве аргументов команды были указаны оси, то сразу после прохождения 0-метки начнётся линейная интерполяция по этим осям аналогично команде G1. В команде G33 задаётся оборотная подача, т. е. через команду F можно задавать

шаг резьбы (см. рисунок 2.4.5). Оборотная подача означает, что скорость перемещения инструмента будет пропорциональна количеству оборотов шпинделя в минуту.

Формат команды:

```
; Синхронизация шпинделя с 0-меткой
G33
```

```
; Нарезание резьбы вдоль оси Z
G33 Z__ F__
```

```
; Выполнение линейной интерполяции (G1) сразу после синхронизации с
   0-меткой
G33 X__ Y__ ... F__
```

Нарезание резьбы обычно повторяют из одной и той же точки, чтобы траектория движения инструмента были одинаковой как при чистовой, так и при черновой обработке. Если в процессе одного из проходов при обработке резьбы траектория изменится, резьба будет испорчена. Поэтому нарезание резьбы начинается в момент, когда шпиндель пройдёт свою 0-метку (выдаст сигнал, что он сделал один полный оборот). При этом скорость шпинделя должна оставаться одинаковой как при повторных проходах, так и в рамках всего прохода. Если скорость шпинделя будет изменяться (несовершенство сервосистемы и т. п.), возможны искажения шага резьбы. Поэтому рекомендуется задать величину перемещения инструмента большей, чем длина резьбы.

Пример. Нарезание резьбы с шагом в 1.2мм.

```
1 S100
2 G33 Z20 F1.2
```

При нарезании резьбы скорость движения инструмента определяется, исходя из количества оборотов шпинделя. Данные о количестве оборотов снимаются с датчика, если он был установлен на шпинделе. В противном случае берётся заданное в программе значение с учётом процентовки. Во время нарезания резьбы дозволяется менять значение процентовки шпинделя, однако, из-за нелинейных характеристик приводов либо переключения скоростного диапазона возможно искажение шага резьбы. Поэтому рекомендуется производить нарезание резьбы с постоянной скоростью шпинделя.

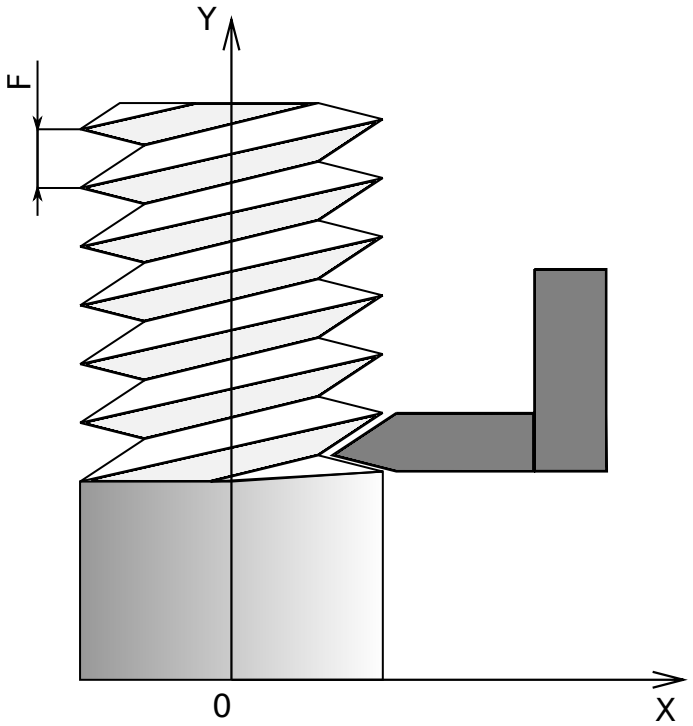


Рис. 2.4.5: Нарезание резьбы через G33

2.4.5. Функция пропуска (G31)

Команда G31 аналогична команде G1 за тем исключением, что её работа может быть прервана внешним сигналом пропуска. При поступлении внешнего сигнала команда прекращает свою работу (не доходя до конечной), и начинается выполнение следующего кадра. Внешний сигнал пропуска может быть послан из программы ПЛК при возникновении какого-либо события.

Данную функцию можно использовать для измерений размеров заготовки, если на инструменте установлен соответствующий датчик.

Формат команды:

```
G31 X__ Y__ ... F__
```

Пример. Измерения высот детали

```
G0 X0 Y0 Z20
G0 Z5 ;подводим инструмент ближе к детали
#A = 0
WHILE [#A < 10] DO
    G91 G0 X#A
    G90 G31 Z-1 F50 ;двигаемся вниз, пока датчик не коснётся поверхности заготовки
    #POS[#A] = Z ;запоминаем координату по Z
    G0 Z5
    #A = #A + 1
DONE
#A = 0
WHILE [#A < 10] DO
    PRINT "X: " #A "Z:" #POS_Z[#A] ;Выводим все полученные точки на экран
DONE
```

2.5 Подача

2.5.1. Общие сведения

Подача представляет из себя скорость движения инструмента. При движении подача раскладывается на составляющие по каждой из осей. Скорость движения инструмента может быть вычислена при помощи формулы 2.5.1:

$$F = \sum_{i=1}^{N} F_i = F_1 + F_2 + \ldots + F_N \tag{2.5.1}$$

Существует два вида подачи: ускоренная подача и подача резания.

- Ускоренная подача предназначена для позиционирования инструмента на максимальной скорости в нужную точку.
- Подача резания используется при обработке деталей и обычно намного ниже ускоренной, чтобы обеспечить большую точность обработки детали. Данный вид подачи задаётся в управляющей программе.

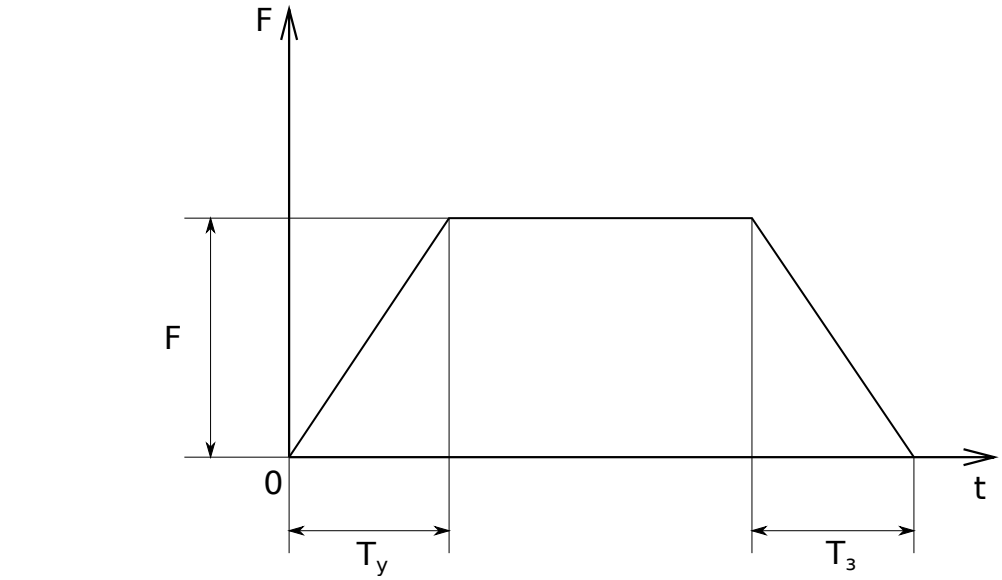
Подача представляет из себя скорость движения инструмента. Но привода либо не могут мгновенно начать движение с заданной скоростью вследствие нелинейных характеристик двигателей (см. рисунок 2.5.1, б), либо будут приводить к жёсткому механическому удару. Поэтому происходит автоматическое ускорение в начале движения и автоматическое замедление в конце

движения. Время ускорения и время замедления представляют из себя довольно маленькие величины и определяются параметрами N6n00 для подачи резания и N6n01 для ускоренной подачи (см. рисунок 2.5.1, а).

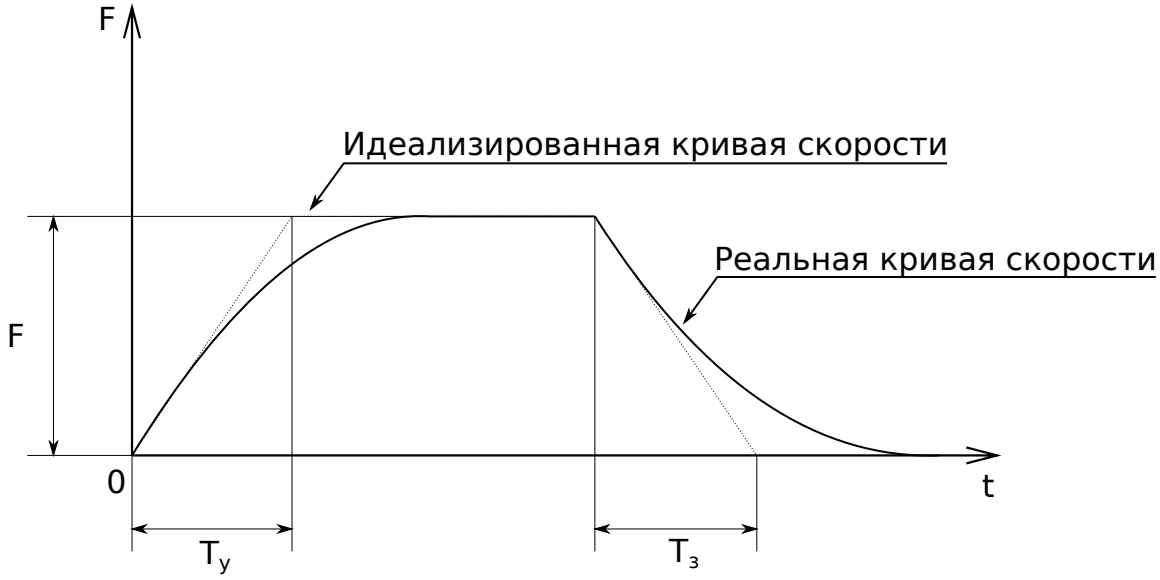
Предусмотрено несколько режимов изменения подачи. При включенной стыковке кадров по скорости (G62, G64), подача изменяется мгновенно, без участков ускорения/замедления. В результате на больших скоростях могут происходить механические удары, если угол смены направления движения приближается к 90 градусам. В результате может происходить незначительное искажение траектории движения инструмента, как показано на рисунке 2.5.2.

Существует два режима задания подачи:

- в милиметрах в минуту (мм/мин), G94;
- в оборотах в минуту (об/мин), G95.



(а)



(б)

Рис. 2.5.1: Идеализированная кривая скорости (а) и реальная кривая скорости (б)

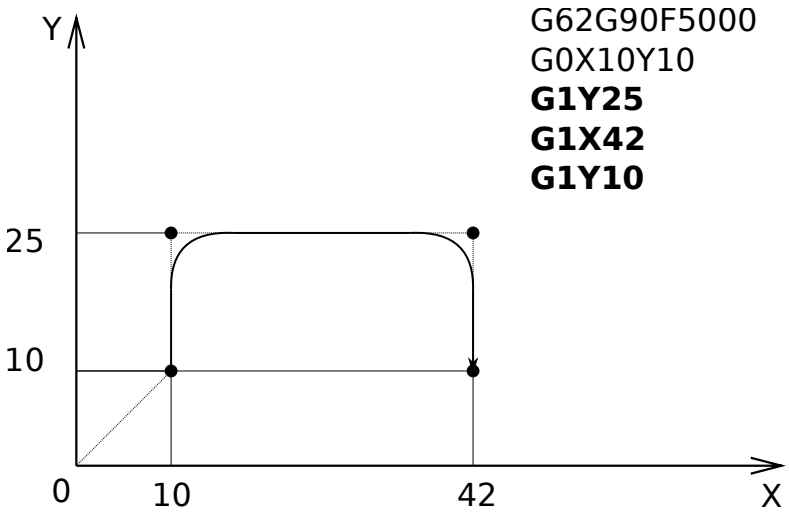


Рис. 2.5.2: Траектория движения при стыковке скоростей

2.5.2. Подача за минуту, G94

При указании флага G94 в том же кадре должна быть задана новая подача параметром F в мм/мин. После этого все последующие кадры будут задаваться с подачей в мм/мин. Режим G94 будет действовать до смены режима флагом G95.

С помощью процентовки, находящейся на пульте оператора, можно применять ручную коррекцию скорости движения инструмента. Но стоит быть осторожным при работе с командами нарезания резьбы и т. п., так как использование коррекции подачи может привести к искажению траектории движения инструмента.

Величина подачи по каждой из осей должна лежать в пределах:

$$0 \leqslant F_i \leqslant F_{i,max}$$

$F_{i,max}$ — максимальное значение подачи по оси (параметр N6n15).

Если заданная подача по какой-либо оси превышает максимальное значение, то общая величина подачи будет уменьшена до такого наиболее близкого к заданному значения, чтобы по всем осям величина подачи не превышала максимальное значение.

2.5.3. Оборотная подача за минуту, G95

При указании флага G95 в том же кадре должна быть задана новая подача параметром F в об/мин. После этого все последующие кадры будут задаваться с подачей в об/мин. Режим G95 будет действовать до смены режима флагом G94.

С помощью процентовки, находящейся на пульте оператора, можно применять ручную коррекцию скорости движения инструмента. Но стоит быть осторожным при работе с командами нарезания резьбы и т. п., так как использование коррекции подачи может привести к искажению траектории движения инструмента.

В режиме подачи в об/мин скорость движения инструмента пропорциональна скорости вращения шпинделя. Дополнительную информацию можно получить в разделе 2.4.4 «Нарезание резьбы».

2.5.4. Управление подачей (G09, G61…G64)

Существует несколько режимов управление скоростью подачи.

- G09 — точный останов в кадре.

Таблица 2.5.1: Команды управления подачей

Команда	Описание
G09	Точный останов в кадре
G61	Режим точного останова
G62	Режим стыковки скоростей
G63	Режим нарезания резьбы
G64	Режим грубой обработки

- G61 — режим точного останова (действует до смены флагами G62, G63, G64). Действие аналогично команде G09, но применяется и к последующим кадрам.
- G62 — режим стыковки скоростей.
- G63 — режим нарезания резьбы.
- G64 — режим грубой обработки.

G09 — точный останов в кадре

Применяется только к кадру, в котором указана.

Инструмент в конечной точке кадра тормозится до нулевой скорости, затем выполняется проверка выхода в заданную позицию по обратной связи. Как только инструмент окажется в нужной точке, начнёт выполняться следующий кадр.

Данный модификатор рекомендуется применять, если необходимо точно выйти в заданную в кадре точку траектории.

Пример. Движение с точным остановом

```
G90 G62 ;включили режим стыковки скоростей
G1 X100 Y100 F2000
G1 X150 Y100
G1 X200 Y100
G9 G1 X250 Y150 ;точный останов в точке (250;150)
G1 X300 Y150
```

G61 — режим точного останова

Действует до смены флагами G62, G63, G64.

Инструмент в конечной точке кадра тормозится до нулевой скорости, затем выполняется проверка выхода в заданную позицию согласно датчи-

кам. Как только инструмент окажется в нужной точке, начнёт выполняться следующий кадр.

Данный режим рекомендуется применять, если необходима повышенная точность изготовления детали, так как в данном режиме инструмент будет двигаться в точности согласно заданной траектории.

Пример. Движение с точным остановом

```
G90 G61 ;включили режим точного останова
G1 X100 Y100 F2000
G1 X150 Y100
G1 X200 Y100
G1 X250 Y150
G1 X300 Y150
```

G62 — режим стыковки скоростей

Действует до смены флагами G61, G63, G64.

Инструмент не выполняет торможение в конце кадра, если угол между текущим кадром и следующим не превышает значение, заданное в параметре N3037. Если угол направления движения между кадрами недопустим ($|\alpha| \geqslant N_{3037}$), то в конце текущего кадра будет происходить полное торможение. Если угол между кадрами допустим ($|\alpha| < N_{3037}$), то скорость в конце текущего кадра (и в начале следующего) вычисляется пропорционально углу:

$$F = \frac{F_a + F_b}{2} \cdot \frac{N_{3037} - |\alpha|}{N_{3037}} \tag{2.5.2}$$

F — подача на границе кадров;

F_a — подача в конце текущего кадра;

F_b — подача в начале следующего кадра;

α — угол между кадрами;

N_{3037} — максимальное значение угла, прописанное в параметре N3037.

Пример. Движение со стыковкой скоростей

```
G90 G62 ;включили режим точного останова
G1 X100 Y100 F2000
G1 X150 Y100
G1 X200 Y100
G1 X250 Y150
G1 X300 Y150
```

G63 — режим нарезания резьбы

Действует до смены флагами G61, G62, G64.

Инструмент не выполняет торможение в конце кадра, если угол между текущим кадром и следующим не превышает значение, заданное в параметре N3038. Если угол направления движения между кадрами допустим, то следующий кадр начнёт выполнение с подачей текущего кадра, даже если у него задана другая подача (в ходе исполнения следующего кадра подача будет опускаться/подниматься до заданной). Иначе происходит торможение до нулевой скорости.

В режим G63 не работает останов подачи и изменение подачи через процентовку.

Изменение процентовки подачи не влияет на скорость движения инструмента.

G64 — режим резания

Действует до смены флагами G61, G62, G63.

Инструмент не выполняет торможение в конце кадра, если угол между текущим кадром и следующим не превышает значение, заданное в параметре N3038. Если угол направления движения между кадрами допустим, то следующий кадр начнёт выполнение с подачей текущего кадра, даже если у него задана другая подача (в ходе исполнения следующего кадра подача будет опускаться/подниматься до заданной). Иначе происходит торможение до нулевой скорости.

2.6 Дополнительные функции

2.6.1. Задержка по времени (G4)

Команда G4 позволяет сделать паузу в отработке программы определённой длительности.

Формат команды:

```
; пауза длительностью X секунд
G4 X__
```

```
; пауза длительностью P микросекунд
G4 P__
```

X__ — длительность паузы в секундах (вещественное число).

P__ — длительность паузы в микросекундах (целое число).

Если не заданы ни X, ни P, то происходит точный останов.

Команду можно использовать для того, чтобы заставить программу приостановиться на какое-то время. Например, можно делать паузы между этапами обработки, чтобы дать инструменту время остыть.

2.7 Возврат в специальные положения

Станки с ЧПУ имеют определённые положения, которые можно считать для них базовыми (референтными). В одном случае это может быть координата, где происходит выход в 0. В другом — позиция, в которой происходит смена инструмента.

2.7.1. G28 — возврат в основное положение

Команда G28 производит перемещение инструмента в основное положение станка. Основное положение может соответствовать «0» станка (если N7n01 = 0) либо определённой позиции, заданной через параметры (N7n01).
Формат команды:

```
; перемещение в основное положение по всем осям
G28
```

```
; перемещение в основное положение по заданным осям
G28 X__ Y__ ...
```

Команда G28 аналогична команде G0. Происходит перемещение на ускоренной подаче в координаты, указанные аргументами команды G28 (промежуточную точку), а затем происходит перемещение в основное положение. Если в команде G28 не были заданы координаты осей, то происходит перемещение либо напрямую в заданную точку, либо по координатам, заданным с помощью команды G28 в программе ранее. Координаты осей, задаваемые в команде G28, сохраняются в системе УЧПУ до конца работы управляющей программы.

Пример. Выход в базовую позицию по G90 (рис. 2.7.1, а)

```
1 G90 G0 X80 Y120
2 G28 X175 Y193 ; перемещение в точку (175, 193), затем в (296, 206)
```

Пример. Выход в базовую позицию по G91 (рис. 2.7.1, б)

```
1 G90 G0 X80 Y120
2 G91 G28 X95 Y73 ; перемещение в точку (175, 193), затем в (296, 206)
```

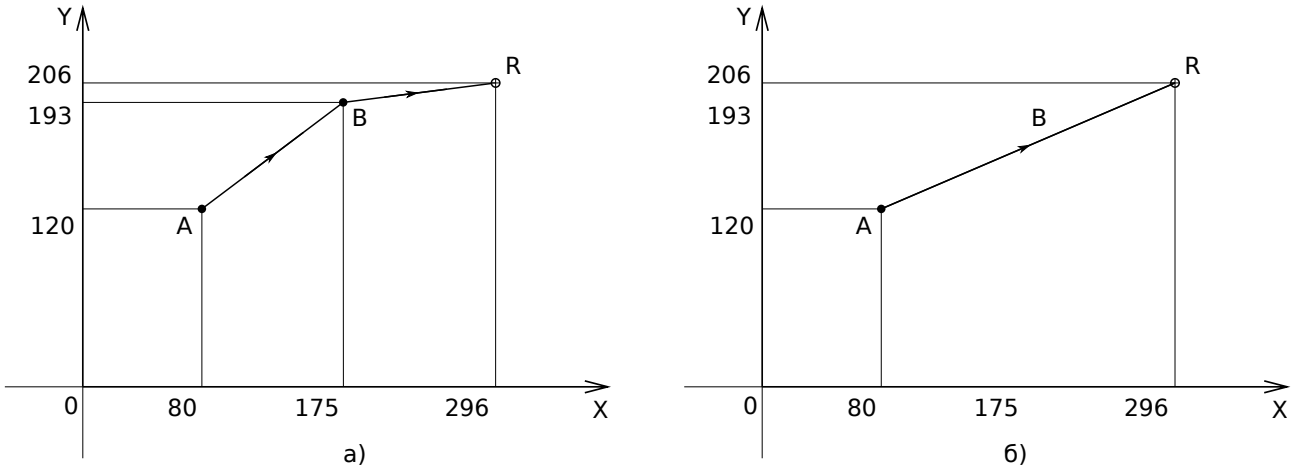


Рис. 2.7.1: Примеры выхода в базовую позицию X=206, Y=296

Для выхода в основную позицию станка сразу (без выхода в промежуточную точку) можно задать режим G91 и нулевые перемещения по осям в команде G28.

Пример. Выход в основную позицию без промежуточной точки (рис. 2.7.1, б)

```
G90 G0 X80 Y120
G91 G28 X0 Y0 ; перемещение в точку (296, 206)
```

Если промежуточная точка в программе всегда одна и та же, то удобно задать её положение при первом вызове G28, а последующие вызывать без задания координат промежуточной точки. В таком случае при первом вызове G28 произойдёт запоминание координат промежуточной точки, а в последующие вызовы G28 будет происходить перемещение в промежуточную точку, запомненную ранее.

Пример. Повторный выход в основную позицию с той же самой промежуточной точкой

```
G90 G0 X80 Y120
G28 X175 Y193 ; перемещение в точку (175, 193), затем в (296, 206)
G0 X0 Y0
G28 ; перемещение в точку (175, 193), затем в (296, 206)
```

2.7.2. G29 — возврат из основной позиции

Функция предназначена для возврата из позиции, в которую был осуществлён переход через команды G28 или G30. Возврат осуществляется в координаты, указанные в команде G29, но через промежуточное положение, указанное ранее в командах G28 или G30.

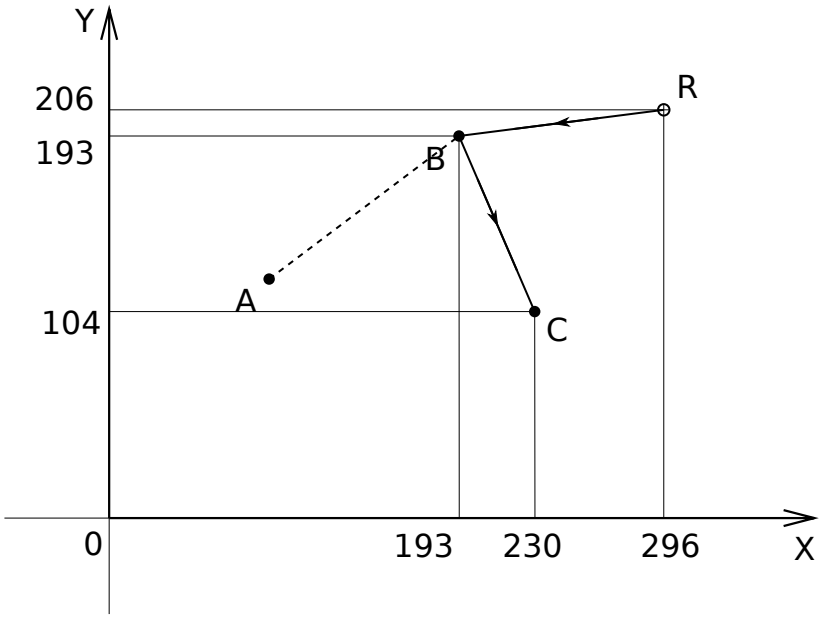


Рис. 2.7.2: Пример возврата из осовного положения

Формат команды:

```
; возврат по всем осям
G29
```

```
; возврат по заданным осям
G29 X__ Y__ ...
```

Если в команде G29 не указаны оси, то происходит возврат в промежуточное положение, запомненное по команде G28 или G30.

Если в команде G29 указаны какие-либо оси, то происходит возврат в промежуточное положение, запомненное по команде G28 или G30, а затем происходит перемещение по осям, заданным в команде G29.

Пример. Возврат из основной позиции по G29 (рис. 2.7.2)

```
1 G90
2 G0 X80 Y120
3 G28 X175 Y193 ; перемещение в точку (175, 193), затем в (296, 206)
4 G29 X230 Y104 ; перемещение в точку (175, 193), затем в (230, 104)
```

2.7.3. G30 — переход в референтную позицию

Команда действует аналогично команде G28, но осуществляет переход во второе, третье и т. д. референтные положения, заданные через параметры (N7n01…N7n09).

Формат команды:

```
; перемещение в референтное положение 2 по всем осям
G30
```

```
; перемещение в референтное положение 2 по заданным осям
G30 X__ Y__ ...
```

```
; перемещение в референтное положение P по всем осям
G30 P__
```

```
; перемещение в референтное положение P по заданным осям
G30 P__ X__ Y__ ...
```

P__ — задаёт номер референтного положения, например, P2, P3, P4 и т.д. Если P = 1, то команда производит перемещение в основное положение (аналогично G28). Если значение P задано меньше 1, либо больше 9, то возникает сообщение об ошибке и выставляется сигнал сброса программы.

Позиция каждого референтного положения задаётся через параметры. Номера параметров для каждого из положений приведены в таблице 2.7.1. Значение координат, записываемое в параметрах, указывается без каких-либо смещений относительно нуля станка.

Если в команде G30 указано перемещение по одной или более осям, то команда переместится сначала в указанные координаты — промежуточное положение (аналогично команде G0), а затем будет произведено перемещение в референтную позицию. Перемещение в промежуточную и референтную позиции будет производиться только по указанным в G30 осям.

Пример. Переход во 2-е реф. положение по осям X и Y через промежуточное положение (X=100; Y=200)

```
1 G90 G30 X100 Y200
```

Если в кадре команды G30 не было указано перемещения ни по одной из осей, то команда G30 произведёт перемещение инструмента в референтное положение по всем осям, у которых выставлен параметр N6n47.

Пример. Переход в 3-е референтное положение по всем осям

```
1 G30 P3
```

Если необходимо произвести перемещение в референтное положение только по определённым осям без выхода в промежуточную точку, то необходимо задать нулевые перемещение по нужным осям в системе координат G91 (т. е. сделать текущее положение промежуточным).

Таблица 2.7.1: Соответствие референтных позиций параметрам

Номер позиции (P)	Параметр	Описание
1	N7n01	Основное положение (G28)
2	N7n02	Референтное положение №2
3	N7n03	Референтное положение №3
4	N7n04	Референтное положение №4
5	N7n05	Референтное положение №5
6	N7n06	Референтное положение №6
7	N7n07	Референтное положение №7
8	N7n08	Референтное положение №8
9	N7n09	Референтное положение №9

Пример. Переход в 4-е референтное положение по оси X (без промежуточного положения)

```
1 G91 G30 P4 X0
```

2.8 Системы координат

Станок имеет набор осей, перемещение каждой из которых может изменять одну или более декартову координату инструмента. Для перемещения инструмента в определённую точку, необходимо знать координаты этой точки по всем осям. На токарных станках обычно используются две оси: X и Z. Соответственно координаты задаются следующим образом:

```
X__ Z__
```

На фрезерном станке может быть 3 оси (или более, если используется поворотный стол или нечто подобное): X__ Y__ Z__. Соответствующие координаты задаются следующим образом:

```
X__ Y__ Z__
```

В стандартных случаях эти оси соответствуют декартовым осям координат, но наладчик станка может это изменять по своему усмотрению.

Координаты можно задавать в нескольких системах координат.

1. Машинные координаты (система координат станка).
2. Рабочие координаты (система координат заготовки).

Таблица 2.8.1: Системы координат заготовки

Команда	Система координат
G53	Система координат станка
G54	Система координат заготовки 1
G55	Система координат заготовки 2
G56	Система координат заготовки 3
G57	Система координат заготовки 4
G58	Система координат заготовки 5
G59	Система координат заготовки 6

3. Относительные координаты (локальная система координат).

У каждого станка есть своя собственная нулевая точка («0» станка). Она служит для начала отсчёта координат, но не обязательно соответствует нулевым координатам. Если нулевая точка соответствует нулевым координатам, то она считается точкой отсчёта системы координат станка. Определение «0» станка выполняется в ручном режиме после включения питания станка с помощью пульта оператора. Данная операция называется выходом в «0». После выхода в «0» координаты нулевой точки запоминаются в ЧПУ до выключения питания.

Помимо системы координат станка существует несколько рабочих систем координат (системы координат заготовки). Можно переходить из одной системы координат в другую при помощи команд G53…G58 (см. рисунок 2.8.1).

Команда G53 выбирает систему координат станка и подходит для перемещения в позицию смены инструмента либо для перемещения в какую-либо определённую точку для выполнения обслуживания. Координаты, задаваемые в системе координат G53, действительны, только если был осуществлён выход в «0» после включения станка, либо если используются датчики абсолютного положения (обычно используются инкрементальные датчики).

2.8.1. Системы координат заготовки (G54…G59)

Система координат заготовки используется при обработке заготовки. Смещение каждой системы координат относительно «0» станка задаётся через интерфейс пользователя либо записывается в соответствующий файл конфигурации вручную (только для опытных пользователей). Команды G54…G59 являются модальными (флагами), т.е. действуют и на последующие кадры тоже.

Пример. Переход между системами координат

```
G90
G54 G0 X0 Y0 Z0
G0 Z10
G1 Z0 F1000
G1 X100
G1 Z10
G55 G0 X0 Y0 Z0
G0 Z50
G1 Z30 F1000
G1 X-100
G1 Z50
G56 G0 X0 Y0 Z0
```

Привязка системы координат заготовки

Для привязки системы координат заготовки из УП можно использовать команду G10.

Формат команды:

```
G10 L2 X__ Y__ ...
```

```
G10 L2 P__ X__ Y__ ...
```

В качестве аргументов передаются координаты по осям, к которым будет привязана текущая координата. Привязка осуществляется только по тем осям, которые указаны в рамках команды.

Параметр P указывает, в какой системе координат заготовки будет осуществляться привязка. Значения и соответствующие им системы координат заготовки указаны в таблице 2.8.2. Если параметр P не указан, то по умолчанию он принимается равным 0.

В режиме G90 текущая координата привязывается к той, которая задана в качестве значения соответствующей оси. В режиме G91 осуществляется смещение системы координат заготовки на величины, заданные в качестве значения осей.

Таблица 2.8.2: Значения параметра P команды G10 L2

Значение P	Система координат заготовки
0	Текущая система координат заготовки
1	G54
2	G55
3	G56
4	G57
5	G58
6	G59

2.8.2. Смещение системы координат (G52)

Систему координат станка можно смещать на заданную величину с помощью команды G52.

Формат команды:

```
G52 X__ Y__ ...
```

В качестве аргументов задаются координаты в текущей системе координат, которые будут соответствовать нулевым координатам новой системы координат.

Команда является модальной, т. е. будет действовать на все последующие кадры программы, пока не будет отменена. Повторное задание смещения системы координат с помощью команды G52 в режиме G90 перезаписывает смещение, заданное ранее. В режиме G91 смещения прибавляются к указанным ранее. Отмена смещения системы координат происходит исполнением команды G52 со смещениями осей, равными нулю.

Пример. Задание новой системы координат

```
1 G90 G54 ;Система координат заготовки 1
2 G52 X0 Y0 ;Отмена предыдущего смещения системы координат по осям X и Y
3 G0 X1223 Y224 ;Переместились в координаты X=1223, Y=224
4 G52 X1223 Y224 ;Сместили систему координат так, что X=0 и Y=0
5 G0 X0 Y0 ;Уже находимся в данной точке, поэтому перемещения не будет
6 G91 G52 X1 Y1 ; Новые координаты: X=-1, Y=-1
7 G90 G0 X0 Y0 ;Переместились на 1 мм по каждой из осей в положительном
    направлении
8 G52 X0 Y0 ;Отмена смещения системы координат
```

2.8.3. Пользовательская система координат (G92)

Помимо заранее определённых систем координат заготовки существует возможность задавать свою собственную систему координат через команду G92.

Формат команды:

```
G92 X__ Y__ ...
```

В качестве аргументов задаются значения координат по осям, которые будут соответствовать текущим координатам новой системы координат. Таким образом данная команда меняет текущие координаты на заданные.

Команда является модальной, т. е. будет действовать на все последующие кадры программы, пока не будет отменена с помощью команды G92.1. Режимы G90 и G91 на команду G92 не воздействуют.

Пример. Задание новой системы координат

```
1 G90 G54 G92.1 ;Система координат заготовки 1
2 G0 X1223 Y224 ;Переместились в координаты X=1223, Y=224
3 G92 X0 Y0 ;Сместили систему координат так, что X=0 и Y=0
4 G0 X0 Y0 ;Уже находимся в данной точке, поэтому перемещения не будет
5 G92 X1 Y1 ;Новые координаты: X=1, Y=1
6 G90 G0 X0 Y0 ;Переместились на 1 мм по каждой из осей в отрицательном
    направлении
7 G92.1 ;Отмена пользовательской системы координат
```

2.8.4. Абсолютная (G90) и относительная (G91) системы координат

G90 — абсолютная система координат

Абсолютная система координат означает задание координат относительно начала текущей системы координат. Флаг действует до конца работы программы, либо до отмены командой G91.

G91 — относительная система координат

Относительная система координат означает задание координат в приращениях, т. е. в расстояниях относительно текущей точки. Данную систему координат удобно использовать при выполнении каких-либо цикличных однотипных операций при обработке детали. Флаг действует до конца работы программы либо до отмены командой G90.

Пример. Сверление канавок вдоль оси X

```
G90 G54 G0 X0 Y0 Z0
G91 S100 M3
G1 Z-10 F100
G1 X1
G1 Z10 F1000
G1 Z-10 F100
G1 X1
G1 Z10 F1000
G1 Z-10 F100
G1 X1
G1 Z10 F1000
```

2.8.5. Масштабирование системы координат (G51)

Масштабирование системы координат задаётся модальной командой G51. Масштабирование осуществляется только по декартовым осям X, Y и Z. В качестве аргументов указывается центр масштабирования системы координат и масштаб в общем виде либо отдельно для каждой оси.

Масштаб по осям X, Y и Z задаётся через аргументы I, J и K соответственно. Общий масштаб задаётся аргументом P. В качестве значения масштаба указывается коэффициент масштабирования. Коэффициент масштабирования, равный 1.0, соответствует оригинальным размерам. Аргументы I, J и K можно указывать совместно с аргументом P: в таком случае аргументы I, J и K домножаются на P.

Отрицательные значения коэффициентов масштабирования равносильны зеркальному отображению системы координат относительно центра масштабирования.

При использовании круговой интерполяции с заданием радиуса центр окружности корректно масштабируется только при указании общего масштаба через аргумент P. При различных значениях маштаба по осям поведение круговой интерполяции будет неопределённым и может измениться в следующих версиях ЧПУ.

Формат команды:

```
G51 P__ ; масштабирование по всем осям

G51 X__ Y__ Z__ ; масштабирование отдельно по каждой оси

G51 X__ Y__ Z__ P__; комбинированное масштабирование по всем осям и
   отдельно по каждой
```

Отмена круговой интерполяции осуществляется командой G50.

2.8.6. Поворот системы координат в плоскости (G68)

Поворот системы координат осуществляется с помощью модальной команды G68. Поворот осуществляется в рамках текущей плоскости интерполяции (G17, G18, G19, G220). При вызове команды G68 необходимо указать координаты точки, вокруг которой будет осуществляться поворот системы координат (центр вращения). Параметр R задаёт угол, на который необходимо повернуть систему координат.

Формат команды:

```
G17 G68 X__ Y__ R__ ;поворот в плоскости XY
```

```
G18 G68 X__ Z__ R__ ;поворот в плоскости XZ
```

```
G19 G68 Y__ Z__ R__ ;поворот в плоскости ZY
```

```
G220 A0 B0
G68 A__ B__ R__ ;поворот в плоскости произвольных осей A и B
```

Отмена поворота системы координат осуществляется с помощью команды G69.

2.8.7. Поворот системы координат через углы Эйлера (G68.2)

Поворот системы координат в пространстве через углы Эйлера осуществляется с помощью команды G68.2. В качестве аргументов задаётся центр вращения и углы Эйлера.

Формат команды:

```
G68.2 X__ Y__ Z__ αI βJ γK
```

Аргументы команды:

- X, Y, Z — координаты центра вращения системы координат;
- α — угол прецессии (поворот вокруг оси Z);
- β — угол нутации (поворот вокруг новой оси X);
- γ — угол собственного вращения (поворот вокруг новой оси Z).

Отмена поворота системы координат осуществляется с помощью команды G69.1.

Ограничения при использовании

- В случае использования совместно с коррекцией на радиус (G41, G42) коррекция на радиус должна начинаться после включения трансформации и выключаться до отмены трансформации.

Пример. Использование поворота с коррекцией на радиус

```
G68.2 X0 Y0 Z0 I45 J45 K0
G41 G0 X10 Y10
// ...
G40 G1 X10 Y10
G69.1
```

- Функцию нельзя использовать совместно с поворотом в плоскости через G68.

2.9 Коррекция на радиус

2.9.1. Команды коррекции (G40...G42)

Коррекция на радиус предполагает изменение кадров таким образом, чтоб учесть радиус инструмента при обработке детали. Существует два типа коррекции на радиус: коррекция слева (G41) по направлению движения и коррекция справа (G42) по направлению движения. Коррекция на радиус означает смещение инструмента относительно задаваемой траектории на величину радиуса перпендикулярно траектории перемещения.

Если G41 поменять на G42, то новая траектория окажется по другую сторону от изначального контура детали. Например, если по G41 обрабатывается внешний контур, то при замене G41 на G42 обрабатываться будет не внешний, а внутренний контур. То же самое с направлением движения. Если для всего обрабатываемого по G41 контура детали поменять направление движения на обратное при условии, что контур детали останется тем же самым, то скорректированный по G41 контур будет уже по другую сторону от контура детали.

Коррекция на радиус отменяется командой G40.

Радиус инструмента можно задавать как в таблице инструментов с помощью интерфейса оператора, так и с помощью корректоров D__, указывая номер корректора из таблицы корректоров. Следует учитывать, что радиус, прописанный в таблице инструментов для текущего инструмента, и радиус, заданный корректором, суммируются.

Пример. Коррекция на радиус слева

```
G90 G54 D2
G0 X0 Y0 Z0
G41 ;Коррекция на радиус слева
G1 X100 F100
G1 Y-100 F100
G40 ;Отмена коррекции на радиус
```

Пример. Коррекция на радиус справа

```
G90 G54 D5
G0 X0 Y0 Z0
G42 ;Коррекция на радиус справа
G1 X50 Y10 F100
G1 X60 Y60 F100
G40 ;Отмена коррекции на радиус
```

2.10 Коррекция на длину

2.10.1. Таблица корректоров (H)

Систему координат можно смещать по какой-либо координате при помощи корректоров на длину инструмента. В таблице инструментов можно задать до 9999 корректоров, каждый из которых может быть смещением по одной из осей. Текущий корректор задаётся при помощи команды H.

Формат:

```
H__
```

После команды H необходимо указать номер корректора.

На индикации в окне суппорта показывается номер текущего корректора. Если одновременно были применены несколько разных корректоров к разным осям, то на индикации будет показан номер корректора, заданного последним.

Если выбор корректора осуществляется в режиме отмены коррекции на длину (G49), то на индикации выбранный корректор будет показан, но применён не будет до тех пор, пока не будет вызвана команда G43 или G44.

2.10.2. Команды коррекции на длину (G43, G44, G49)

G49 — отмена коррекции на длину

Формат:

```
G49 ;отмена коррекции на длину
```

В режиме G49 коррекция на длину не применяется. Команда G49 отменяет действие команд G43 и G44. Если до вызова команды G49 был выбран какой-либо H-корректор, то он остаётся текущим на индикации, но его значение сможет быть применено только после вызова команды G43 или G44.

G43 — коррекция на длину в плюс

Формат:

```
G43 ;разрешение коррекции на длину в плюс
```

```
G43 H__ ;коррекция на длину в плюс по декартовой оси Z
```

Команда G43 разрешает коррекцию инструмента в положительном направлении, т. е. значение корректора вычитается из координаты инструмента. Значение корректора берётся из таблицы корректоров по номеру, который указывается в команде H. Коррекция будет применяться к 3-й декартовой оси (традиционно ось Z).

Пример. Коррекция на длину в плюс по оси Z

```
1 G0 G90
2 ;значение корректора H4 = 20
3 G43 H4 Z0 ;перемещение в абсолютную координату 20 (0+20=20)
4 G49 Z0 ;перемещение в абсолютную координату 0
```

Пример. Коррекция на длину в плюс по оси Z с заранее заданным корректором

```
1 G0 G90
2 H3
3 ;значение корректора H3 = 15
4 G43 Z100 ;перемещение в абсолютную координату Z=115 (100+15=115)
5 G49 Z0 ;перемещение в абсолютную координату Z=0
```

G44 — коррекция на длину в минус

Формат:

```
G44 ;разрешение коррекции на длину в минус
```

```
G44 H__ ;коррекция на длину в минус по декартовой оси Z
```

Команда G44 разрешает коррекцию инструмента в отрицательном направлении, т. е. значение корректора прибавляется к координате инструмента. Значение корректора берётся из таблицы корректоров по номеру, который указывается в команде H. Коррекция будет применяться к 3-й декартовой оси (традиционно, ось Z).

Пример. Коррекция на длину в минус по оси Z

```
G0 G90 G53
;значение корректора H5 = 13.5
G43 H5 Z10 ;перемещение в абсолютную координату Y=-3.5 (10-13.5=-3.5)
G49 Z10 ;перемещение в абсолютную координату Y=10
```

Пример. Коррекция на длину в минус по оси Z с заранее заданным корректором

```
G0 G90 G53
;значение корректора H23 = 25
H23
G43 Z25 ;перемещение в абсолютную координату Z=0 (25-25=0)
G49 Z25 ;перемещение в абсолютную координату Z=25
```

2.11 Работа со шпинделем

Шпиндель может работать в двух режимах: в режиме шпинделя и в режиме оси. **В режиме шпинделя** можно задавать скорость вращения шпинделя и включать режим оборотной подачи. **В режиме оси** шпиндель выступает в роли обычной оси, которая может быть интерполирована совместно с другими осями. Режимы работы шпинделя можно переключать между собой.

Для переключения шпинделя в режим оси необходимо указать перемещение по оси, название которой указано в параметрах шпинделя (N5n00). Обычно шпиндель обозначают буквой C.

Для перехода обратно в режим шпинделя необходимо указать скорость вращения шпинделя через S-функцию.

2.11.1. Управление скоростью шпинделя

Управление скоростью вращения шпинделя осуществляется с помощью **S**-функции. S-функция задаёт скорость вращения шпинделя в об/мин либо в м/мин.

Скорость шпинделя регулируется двумя режимами: режим постоянства скорости вращения (G97) и режим постоянства скорости резания (G96).

В фрезерных станках рекомендуется использовать только режим работы G97.

На скорость вращения шпинделя влияет процентовка шпинделя, находящаяся на пульте оператора. С помощью процентовки можно задавать скорость вращения шпинделя в процентах от заданной в программе скорости.

Перед исполнением S-команды происходит полное торможение подачи суппорта аналогично режиму G61. Этот факт необходимо учитывать при разработке управляющих программ.

Возможно ограничить скорость шпинделя из управляющей программы, задав максимальное значение в об/мин с помощью команды **LIMS**.

Формат:

```
LIMS__ ;задать ограничение
```

```
S__ LIMS__ ;установить скорость и задать её ограничение
```

Команда LIMS так же, как и S-команда, приводит к полному торможению подачи суппорта перед исполнением данной команды.

2.11.2. Контроль постоянства скорости вращения

Режим G97 отменяет режим G96.

Формат:

```
G97 S__ ;об/мин
```

```
S__ ;в зависимости от ранее выставленных G96 или G97
```

При включенном режиме G97 S-функция задаёт скорость вращения шпинделя в об/мин. При этом скорость движения инструмента относительно детали будет меняться при изменении расстояния инструмента до оси вращения шпинделя, угловая же скорость шпинделя будет оставаться постоянной. При обработке детали с постоянными оборотами шпинделя с продвижением вглубь будет увеличиваться нагрузка на резец, и, соответственно, может меняться качество поверхности обрабатываемой детали.

В токарных станках данный режим рекомендуется использовать, если

скорость вращения шпинделя не должна меняться при движении инструмента, например, в случае сверления отверстий по центру детали либо при многократных перемещениях на быстром ходу.

Пример. Сверление отверстия с частотой 2000 об/мин

```
G54 G0 X0 Y0 Z10 ;переход в исходную позицию
G97 M3 S2000 F100 ;задаём скорость сверления и подачу
G1 Z-10
G1 Z0
G1 Z-20
G1 Z0
G1 Z-30
G1 Z0
G0 Z10 ;отверстие просверлено, поднимаем фрезу
M5
```

2.11.3. Контроль постоянства скорости резания

Режим G96 отменяет режим G97.

Формат:

```
G96 S__ ;м/мин
```

```
S__ ;в зависимости от ранее выставленных G96 или G97
```

При включенном режиме G96 S-функция задаёт линейную скорость вращения шпинделя в м/мин. Данный режим называется контролем постоянства скорости резания. Если задать линейную скорость, то при изменении положения инструмента относительно детали угловая скорость вращения шпинделя (об/мин) меняется так, чтобы скорость движения инструмента относительно детали оставалась постоянной. Таким образом, нагрузка на резец и деталь будет оставаться постоянной.

Рекомендуется задавать ограничение скорости шпинделя в режиме G96 (контроль постоянства скорости резания). Если ограничение не задавать, то при приближении к оси вращения скорость вращения шпинделя будет значительно возрастать. Позиция инструмента по центру оси вращения шпинделя будет соответствовать максимальной скорости вращения шпинделя (ограниченной либо командой LIMS, либо параметрами станка, либо модулем электроавтоматики станка).

Данный режим не имеет никакого смысла при использовании в фрезерных станках.

Пример. Обтачивание фаски со скоростью 200 м/мин

```
G54
G0 X0 Z10 ;переход в исходную позицию
;задаём скорость вращения в метрах в минуту и подачу
G96 M3 S200 F100 LIMS2000 ;максимальная скорость шпинделя — 2000 об/мин
G1 X-10 Z5 ;снимаем фаску
G0 Z10 ;фаска снята, поднимаем инструмент
M5
G0 X0 ;выход в исходную позицию
```

2.12 Работа с инструментами

Любой инструмент имеет свой собственный набор параметров (высота, радиус, угол наклона и т.п.), поэтому для правильной работы управляющих программ необходимо указывать для каждого инструмента данные параметры. Помимо этого, можно создавать различные привязки инструмента. Привязка инструмента представляет из себя смещение системы координат на величины, прописанные в таблице инструментов для заданной привязки.

2.12.1. Смена инструмента/привязки (T-функция)

Стандартная схема **смены инструмента** может происходить тремя способами (в зависимости от станка):

1. с помощью комбинации T-команды и команды M6 в одной строке;
2. с помощью одной только T-команды;
3. вручную с пульта оператора.

Указание того, какой инструмент необходимо выбрать, осуществляется с помощью **T**-функции. В качестве параметров передаётся номер инструмента и номер привязки.
Формат:

```
T__ ;с указанием номера инструмента
```

```
T__.__ ;с указанием номера инструмента и номера привязки
```

```
T.__ ;с указанием номера привязки для текущего инструмента
```

Если указывается только номер инструмента, то выбор привязки осуществляет модуль ПЛК. Если указывается только номер привязки, то происходит

смещение системы координат для текущего инструмента согласно данным, указанным в привязке.

Перед исполнением T-команды происходит полное торможение подачи аналогично режиму G61.

Примечание. Смену инструмента осуществляет модуль ПЛК. Он вправе выбрать инструмент либо привязку, отличные от заданных в программе.

Пример. Смена инструмента в основной позиции

```
G91 G28 X0 Y0 Z0 ;переход в позицию смены инструмента
G90 G54
T7.4 M6 ;выбираем 7-й инструмент, 4-ю привязку
G27 X0 Y0 Z0 ;переход в нулевую позицию для данной привязки
```

2.13 Вспомогательные функции (M-команды)

В управляющей программе можно указывать команды, которые выполняют специфичные действия либо передают управление модулю ПЛК для осуществления каких-либо операций.

Перед исполнением M-команд происходит полное торможение подачи аналогично режиму G61.

ВНИМАНИЕ! Действия, выполняемые M-командами, и порядок их вызова задаются параметрами и модулем ПЛК. Поэтому их поведение может отличаться от описанного в данном руководстве при использовании нестандартных модулей ПЛК.

2.13.1. Общие команды (M0, M1, M30)

Останов программы (M0)

Команда **M0** выполняет **останов программы** с возможностью последующего возобновления работы программы через кнопку старта. После выполнения данной команды программа должна перейти в состояние паузы.

Команду можно использовать в тех местах управляющей программы, где необходимо произвести ручную смену инструмента, смену заготовки и т.п.

Пример. Сверление отверстия с ручным охлаждением инструмента

```
G0 X0 Y0 Z10
M3 S5000 F100 ;включаем шпиндель
G1 Z-20
G1 Z0 F300
G0 Z10
M0 ;программа останавливается, охлаждаем сверло жидкостью
G0 Z0
G1 Z-40 F100
G1 Z0 F300
G0 Z10
M0 ;программа останавливается, охлаждаем сверло жидкостью
```

Условный останов программы (M1)

Команда **M1** выполняет останов программы аналогично команде M0, если модуль ПЛК разрешил **условный останов**.

Команду можно использовать при отладке управляющей программы.

Повтор программы (M30)

Команда **M30** осуществляет **сброс** управляющей программы и даёт команду **на повторное исполнение** программы **по кнопке старта**.

Если после команды M30 была запущена на исполнение другая управляющая программа, то повтор программы автоматически отменяется.

Пример. Использование команды

```
G65 P1422 X10 Y10 A-5 B5 ;выполняем пользовательскую функцию
M30 ;указываем, что оператор может повторить исполнение программы,
   например, после смены заготовки
G1 X10 ;данная команда исполняться не будет, т.к. вызов M30 завершил
   работу программы
```

2.13.2. Команды управления шпинделем (M3, M4, M5)

Включение шпинделя (M3, M4)

Команда **M3** осуществляет подготовительные действия для **включения шпинделя** и разрешает его вращение в направлении **по часовой стрелке** (в «+»).

Команда **M4** осуществляет подготовительные действия для **включения шпинделя** и разрешает его вращение в направлении **против часовой**

стрелки (в «-»).

Выключение шпинделя (M5)

Команда **M5** осуществляет подготовительные действия для **выключения шпинделя** и запрещает его вращение.
Пример. Включение шпинделя на 10с

```
M3 S1000 ;включили шпиндель
G4 X10 ;ждём 10 секунд
M5 ;выключили шпиндель
```

Смена инструмента (M6)

Команда M6 участвует в смене инструмента. Действие, осуществляемое командой, задаётся отдельно под каждую модель станка.
Пример. Смена инструмента

```
G0 X0 Z0 ;переход в позицию смены инструмента
T4.1 M6 ;смена инструмента
```

2.14 Токарные циклы

2.14.1. Циклы съёма материала (G70, G71, G72,G73)

G70 — цикл чистового съёма материала

Формат (полный):

```
G70 P__ Q__ E__ U__ W__
```

Формат:

```
G70 P__ Q__ ;траектория не изменяется
```

```
G70 P__ Q__ E1 ;траектория изменяется, карманы не срезаются
```

Аргументы команды **G70**

P — метка начала контура профиля.

Q — метка конца контура профиля.

E — корректировать ли траекторию (0 — без коррекции, 1 — инструмент не заходит в карманы), необязательный параметр.

U — допуск по оси X (Δa_x), необязательный параметр.

W — допуск по оси Z (Δa_z), необязательный параметр.

Цикл чистовой обработки предназначен для съёма оставшегося слоя материала с детали, траектория профиля которой задана между кадрами P и Q. Если в цикле параметр E задан в значение 1, то цикл не будет вырезать внутренности карманов. Если параметр E не задан, либо задан в значение 0, то цикл обходит траекторию в её оригинальном виде, либо со смещениями Δa_x и Δa_z, если таковые были заданы.

G71 — многопроходный цикл продольного съёма

Формат (полный):

```
G71 U__ R__ ;первый блок команды
G71 P__ Q__ U__ W__ F__ S__ ;второй блок команды
```

Аргументы **любого блока** команды **G71**

F — подача, используемая для съёма материала F, необязательный параметр.

S — скорость шпинделя, используемая для съёма материала S, необязательный параметр.

Аргументы **первого блока** команды **G71**

U — глубина резания Δd_x (высота срезаемого слоя) по оси X.

R — величина отскока r (расстояние по осям X и Z, на которое инструмент будет отходить под углом 45 градусов после среза очередного слоя материала).

Аргументы **второго блока** команды **G71**

P — метка начала контура профиля.

Q — метка конца контура профиля.

U — допуск по оси X (Δa_x).

W — допуск по оси Z (Δa_z).

Если в цикле подача не задана, то будет использована текущая подача, заданная ранее в программе.

Общая схема многопроходного съёма материала приведена на рисунке 2.14.1.

Начальной точкой цикла считается позиция, в которой инструмент находился в момент вызова цикла.

В каждом проходе цикл выполняет следующие действия:

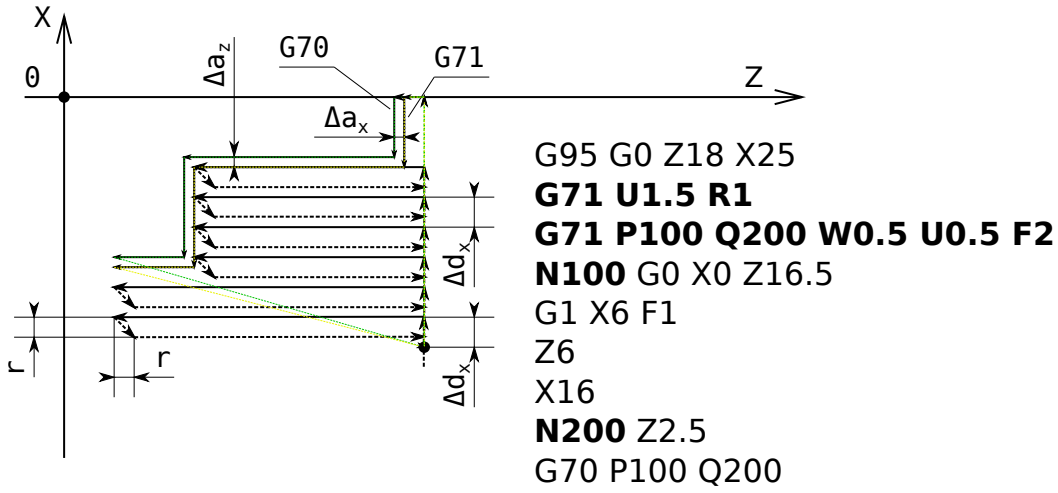


Рис. 2.14.1: Условная схема съёма материала через G71

1. Инструмент движется по оси X на величину глубины резания Δd_x с подачей F из начальной точки цикла.

2. Инструмент движется по оси Z, пока не дойдёт до контура траектории (с учётом допуска Δa_z), заданной между кадрами P и Q.

3. Инструмент отскакивает по осям X и Z на величину r под углом 45 градусов.

4. Инструмент на быстром ходу возвращается по координате Z в начальную точку.

Весь цикл операций выполняется до тех пор, пока инструмент не дойдёт по оси X до конечной точки траектории (с учётом допуска Δa_x). После того, как будут срезаны слои материала, цикл сделает черновой съём оставшегося материала по траектории, оставив слой материала, заданный допусками по осям X и Z (Δa_x и Δa_z соответственно).

G72 — многопроходный цикл поперечного съёма

Формат (полный):

```
G72 W__ R__ ;первый блок команды
G72 P__ Q__ U__ W__ F__ S__ ;второй блок команды
```

Аргументы **любого блока** команды **G72**

F — подача, используемая для съёма материала F, необязательный параметр.

S — скорость шпинделя, используемая для съёма материала S, необязательный параметр.

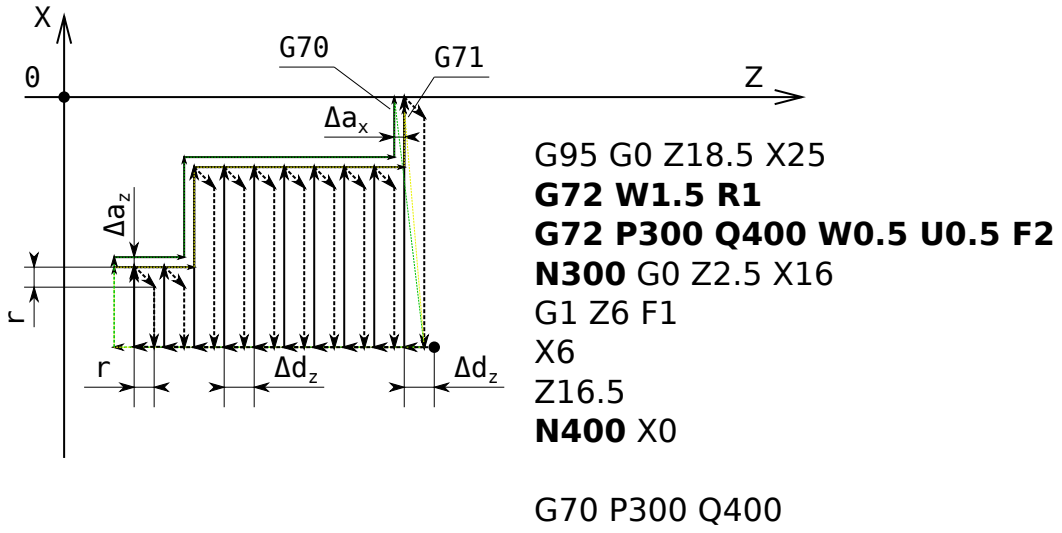


Рис. 2.14.2: Условная схема съёма материала через G72

Аргументы **первого блока** команды **G72**

W — глубина резания Δd_z (ширина срезаемого слоя) по оси Z.

R — величина отскока r (расстояние по осям X и Z, на которое инструмент будет отходить под углом 45 градусов после среза очередного слоя материала).

Аргументы **второго блока** команды **G72**

P — метка начала контура профиля.

Q — метка конца контура профиля.

U — допуск по оси X (Δa_x).

W — допуск по оси Z (Δa_z).

Для задания траектории используются метки N__. Траектория задаётся от метки начала контура до метки конца контура. Началу траектории должен соответствовать кадр G0, осуществляющий позиционирование в начало профиля. Для правильного исполнения цикла начало профиля должно находиться дальше по координате Z от начальной точки цикла (точки возврата), нежели конец траектории.

Если в цикле подача не задана, то будет использована текущая подача, заданная ранее в программе.

Общая схема многопроходного съёма материала приведена на рисунке 2.14.2.

Начальной точкой цикла считается позиция, в которой инструмент находился в момент вызова цикла.

В каждом проходе цикл выполняет следующие действия:

1. Инструмент движется по оси Z на величину глубины резания Δd_z с подачей F из начальной точки цикла.
2. Инструмент движется по оси X, пока не дойдёт до контура траектории (с учётом допуска), заданной между кадрами P и Q.
3. Инструмент отскакивает по осям X и Z на величину r под углом 45 градусов.
4. Инструмент на быстром ходу возвращается по координате X в начальную точку.

Весь цикл операций выполняется до тех пор, пока инструмент не дойдёт по оси Z до конечной точки траектории. После того, как будут срезаны слои материала, цикл сделает черновой проход по траектории, срезав остатки материала, но оставив заданный по осям X и Z допуск.

G73 — многопроходный цикл послойного съёма

Формат (полный):

```
G73 U__ W__ R__ ;первый блок команды
G73 P__ Q__ U__ W__ F__ S__ ;второй блок команды
```

Аргументы **любого блока** команды **G73**

F — подача, используемая для съёма материала F, необязательный параметр.

S — скорость шпинделя, используемая для съёма материала S, необязательный параметр.

Аргументы **первого блока** команды **G73**

U — высота срезаемого в каждом проходе слоя Δd_x по оси X;

W — ширина срезаемого в каждом проходе слоя Δd_z по оси Z;

R — количество проходов N

Аргументы **второго блока** команды **G73**

P — метка начала контура профиля.

Q — метка конца контура профиля.

U — допуск по оси X (Δa_x).

W — допуск по оси Z (Δa_z).

Начальной точкой цикла считается позиция, в которой инструмент находился в момент вызова цикла.

Цикл осуществляет N проходов. В каждом проходе цикл выполняет следующие действия:

1. Инструмент движется к началу траектории, смещённой относительно исходной по оси X на величину $(N-n)\Delta d_x + \Delta a_x$, по оси Z на величину $(N-n)\Delta d_z + \Delta a_z$, где n — порядковый номер текущего прохода.
2. Инструмент перемещается по смещённой траектории до её конца с подачей F.
3. Инструмент возвращается в начальную позицию цикла.

2.14.2. Циклы расточки (G74, G75)

G74 — цикл горизонтальной расточки

Формат (полный):

```
G74 R_
G74 X_ Z_ P_ Q_ R_ F_ S_
```

Аргументы **первого блока** команды **G74**

R — отскок по оси Z, необязательный параметр.

Аргументы **второго блока** команды **G74**

X — конечная координата по оси X, необязательный параметр.

Z — конечная координаты по оси Z.

P — шаг по оси X, необязательный параметр.

Q — шаг по оси Z.

R — отскок по оси X, необязательный параметр.

F — подача, используемая для съёма материала F, необязательный параметр.

S — скорость шпинделя, используемая для съёма материала S, необязательный параметр.

Описание цикла:

Начальной точкой цикла считается позиция (Zn,Xn), в которой инструмент находился в момент вызова цикла $Zt = Zn$,$Xt = Xn$.

1. Движемся по оси Z в координату, вычисленную по формуле $Zt = Zt + Q$, на рабочем ходу.
2. Движемся по оси Z в координату, вычисленную по формуле $Zt - R$, на рабочем ходу (если задан аргумент R).
3. Движение по оси Z в координату, заданную аргументом Z, осуществляется по пунктам 1-2 до тех пор, пока текущая координата Zt не станет равна аргументу Z.
4. Движемся по оси X в координату, вычисленную по формуле $Xt - R$, на рабочем ходу (если задан аргумент R во втором блоке команды).
5. Движемся по оси Z в координату Zn на быстром ходу.
6. Движемся по оси X в координату, вычисленную по формуле $Xt = Xt + P$, на рабочем ходу (если задан аргумент X).
7. Выполняются пункты 1-6 до тех пор, пока Xt не станет равна аргументу X.
8. Делается контрольный проход по пунктам 1-5.
9. Движемся по оси X в координату Xn на быстром ходу.

Схематическое обозначение цикла горизонтальной расточки G74 изображено на рисунке 2.14.3.

G75 — цикл вертикальной расточки

Формат (полный):

```
G75 R_
G75 X_ Z_ P_ Q_ R_ F_ S_
```

Аргументы **первого блока** команды **G75**

R — отскок по оси Z, необязательный параметр.

Аргументы **второго блока** команды **G75**

X — конечная координата по оси X.

Z — конечная координаты по оси Z, необязательный параметр.

P — шаг по оси X.

Q — шаг по оси Z, необязательный параметр.

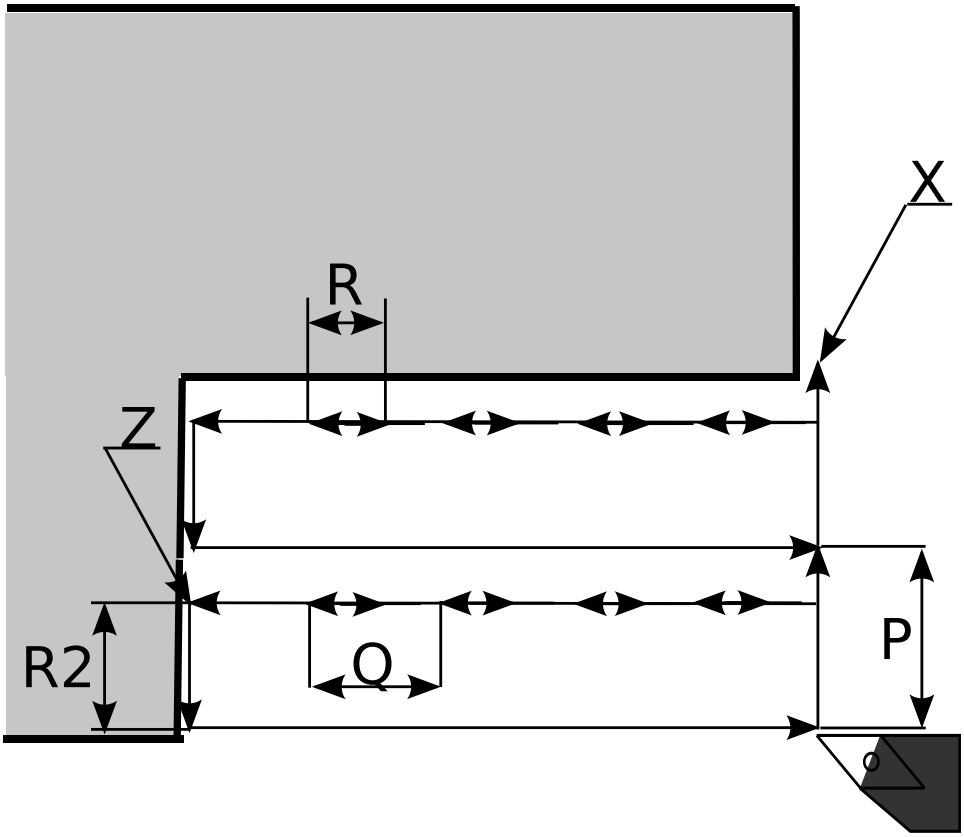


Рис. 2.14.3: Цикл горизонтальной расточки G74

R — отскок по оси X, необязательный параметр.

F — подача, используемая для съёма материала F, необязательный параметр.

S — скорость шпинделя, используемая для съёма материала S, необязательный параметр.

Описание цикла:

Начальной точкой цикла считается позиция (Zn,Xn), в которой инструмент находился в момент вызова цикла $Zt = Zn$,$Xt = Xn$.

1. Движемся по оси X в координату, вычисленную по формуле $Xt = Xt + P$, на рабочем ходу.
2. Движемся по оси X в координату, вычисленную по формуле $Xt - R$, на рабочем ходу (если задан аргумент R во втором блоке команды).
3. Движение по оси X в координату, заданную аргументом X, осуществляется по пунктам 1-2 до тех пор, пока текущая координата Xt не станет равна аргументу X.
4. Движемся по оси Z в координату, вычисленную по формуле $Zt - R$, на рабочем ходу (если задан аргумент R) .

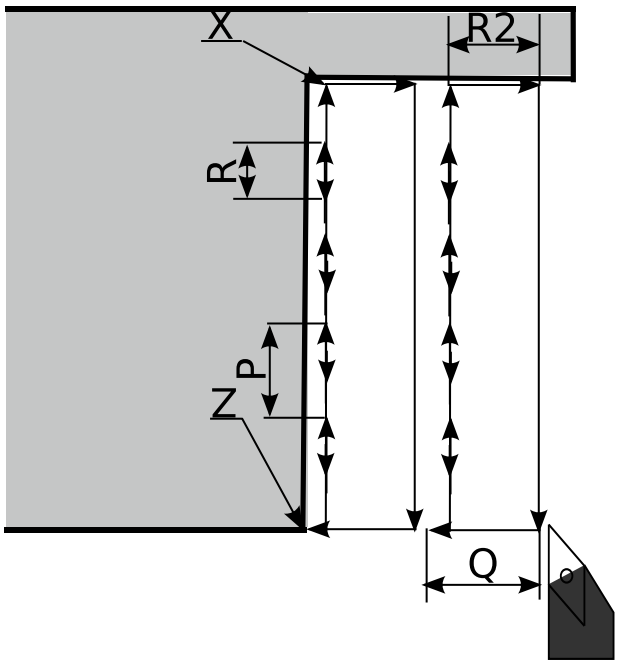


Рис. 2.14.4: Цикл вертикальной расточки G75

5. Движемся по оси X в координату Xn на быстром ходу.
6. Движемся по оси Z в координату, вычисленную по формуле $Zt = Zt + P$, на рабочем ходу (если задан аргумент Z).
7. Выполняются пункты 1-6 до тех пор, пока Zt не станет равна аргументу Z.
8. Делается контрольный проход по пунктам 1-5.
9. Движемся по оси Z в координату Zn на быстром ходу.

Схематическое обозначение цикла вертикальной расточки G75 изображено на рисунке2.14.4

2.14.3. Многопроходный цикл нарезания резьбы (G76)

Формат (полный):

```
G76 P_______ Q__ R__ OA__ ; первый блок команды
G76 X{U}__ Z{W}__ P__ Q__ R__ TT__ TE__ F__ C__ ;второй блок команды
```

```
G76 LC__ OD__ TA__ Q__ R__ OA__ ; первый блок команды
G76 X{U}__ Z{W}__ P__ Q__ R__ TT__ TE__ F__ C__ ;второй блок команды
```

Аргументы **первого блока** команды **G76**

LC — число чистовых проходов n после нарезания резьбы, может быть задано параметром N3402.

OD — расстояние r выхода инструмента из резьбы после её нарезания под углом 45 градусов, может быть задано параметром N3401.

TA — угол инструмента a (отвечает за угол входа в резьбу по умолчанию), может быть задан в таблице инструментов для текущего инструмента.

Pnnrraa — целое число, состоящее из 6 цифр (nnrraa), необязательный аргумент (может быть заменён параметрами LC, OD и TA). Каждые две цифры обозначают отдельный параметр:

- nn — аналогичен аргументу **LC**;
- rr — аналогичен аргументу **OD**;
- aa — аналогичен аргументу **TA**.

Q — глубина обычного прохода нарезания резьбы (радиусное значение), мм. Необязательный аргумент, может быть задан параметром N3403. Может не соответствовать глубине очередного прохода, если выбран алгоритм, отличный от стандартного (см. аргумент TE второго блока).

R — глубина последнего прохода нарезания резьбы (радиусное значение), мм. Необязательный аргумент, может быть задан параметром N3402.

OA — угол выхода из резьбы. Необязательный параметр (значение по умолчанию: 45 градусов).

Аргументы **второго блока** команды **G76**

X(U) — координата окончания резьбы по декартовой оси X (название оси и аргумента может не совпадать), точка D, мм.

Z(W) — координата окончания резьбы по декартовой оси Z (название оси и аргумента может не совпадать), точка D, мм.

P — глубина резьбы h (радиусное значение), мм.

Q — глубина первого прохода нарезания резьбы (радиусное значение), мм. Необязательный аргумент.

R — разница ΔR между начальным и конечным радиусами резьбы, мм. Можно указывать отрицательное значение. Используется для нарезания конической резьбы. Необязательный аргумент, значение по умолчанию: $\Delta R = 0$.

TTlcr — целое число из 3 цифр, задаёт варианты входа резца в резьбу. Необязательный параметр (значение по умолчанию: 001). Возможные значения:

- 001 — вход в резьбу под углом справа, нагрузка на левую кромку резца (по умолчанию);
- 100 — вход в резьбу под углом слева, нагрузка на правую кромку резца;
- 101 — вход в резьбу под углом по очереди справа и слева, а нагрузка, соответственно, по очереди на левую и правую кромку резца;
- 010 — вход в резьбу по центру с нагрузкой на обе кромки резца;
- 111 — вход в резьбу по центру с нагрузкой по очереди на левую и правую кромки резца.

TE — выбор алгоритма вычисления глубины резьбы очередного прохода. Необязательный параметр (значение по умолчанию: 1). Возможные значения:

- 0 — стандартный алгоритм: все проходы, кроме первого, идут с обычной глубиной;
- 1 — алгоритм снижения нагрузки на резец: плавное изменение глубины между величинами первого прохода и обычного прохода (величина очередного прохода не может стать меньше величины обычного прохода), см. рисунок 2.14.6.

C — угол начала резьбы (позиция шпинделя относительно 0-метки, с которой необходимо начинать проход резьбы). Необязательный параметр.

F — шаг нарезания резьбы (оборотная подача), мм/об.

Схематическое обозначение проходов нарезания резьбы показано на рисунке 2.14.5. Схема врезания в резьбу на примере одного витка показана на рисунке 2.14.6.

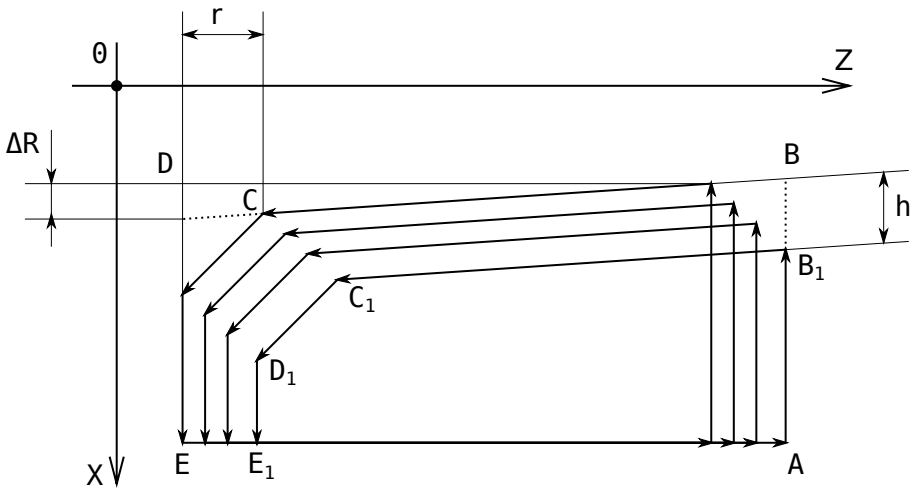


Рис. 2.14.5: Условная схема нарезания резьбы G76

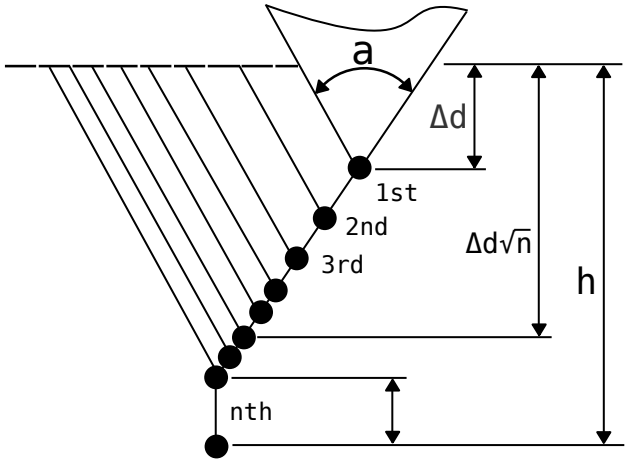


Рис. 2.14.6: Условная схема входа в резьбу по циклу G76

Пример нарезания внутренней резьбы с разными вариантами врезания резца (рис. 2.14.7)

```
G0 X0 Z5
;врезание справа
G91
G76 P010060 Q0.1 R0.05 OD0.5
G76 X2 Z-5 P0.5 Q0.2 F2 TT001
;вход слева
G91 X4
G76 P010060 Q0.2 R0.05 OD0.5
G76 X2 Z-5 P0.5 Q0.2 F2 TT100
;вход по очереди справа и слева
G91 X4
G76 P010060 Q0.2 R0.05 OD0.5
G76 X2 Z-5 P0.5 Q0.2 F2 TT101
;врезание по центру
G91 X4
G76 P010060 Q0.2 R0.05 OD0.5
G76 X2 Z-5 P0.5 Q0.2 F2 TT010
;врезание по центру с поочередным смещением
```

Пример нарезания внутренней и наружной резьбы в двух вариантах (рис. 2.14.8)

```
;внутренняя резьба с движением по Z в минус
G0 X0.5 Z-0.25
G76 P010060 Q0.1 R0.05 OD0.5
G76 X2 Z-5 P0.5 Q0.2 F2
;наружная резьба с движением по Z в минус
G0 X-0.5 Z-0.25
G76 P010060 Q0.1 R0.05 OD0.5
G76 X-2 Z-5 P0.5 Q0.2 F2
;внутренняя резьба с движением по Z в плюс
G0 X0.5 Z0.25
G76 P010060 Q0.1 R0.05 OD0.5
G76 X2 Z5 P0.5 Q0.2 F2
;наружная резьба с движением по Z в плюс
G0 X-0.5 Z0.25
G76 P010060 Q0.1 R0.05 OD0.5
G76 X-2 Z5 P0.5 Q0.2 F2
```

Пример нарезания конусной резьбы (рис. 2.14.9)

```
;Нарезание по оси Z в минус
G0 X0.5 Z5
G76 P010060 Q0.1 OD0.2 R0.05
G76 X2 Z0 P0.5 Q0.2 F2 R0.5
;Нарезание по оси Z в плюс
G0 X5 Z5
G76 P010060 Q0.1 OD0.3 R0.05
G76 X8 Z0 P0.5 Q0.2 F2 R-0.5
```

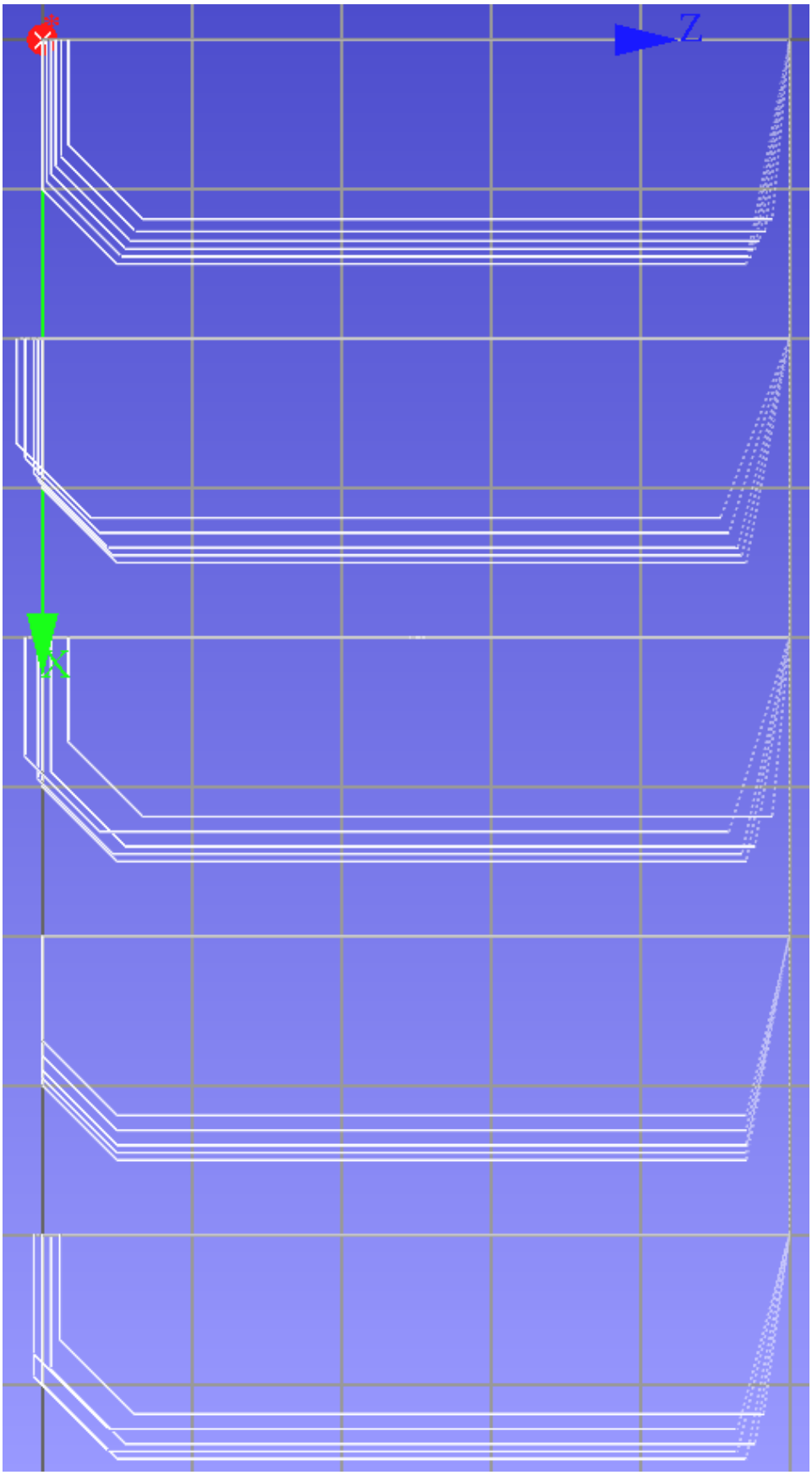


Рис. 2.14.7: Демонстрация врезания резца

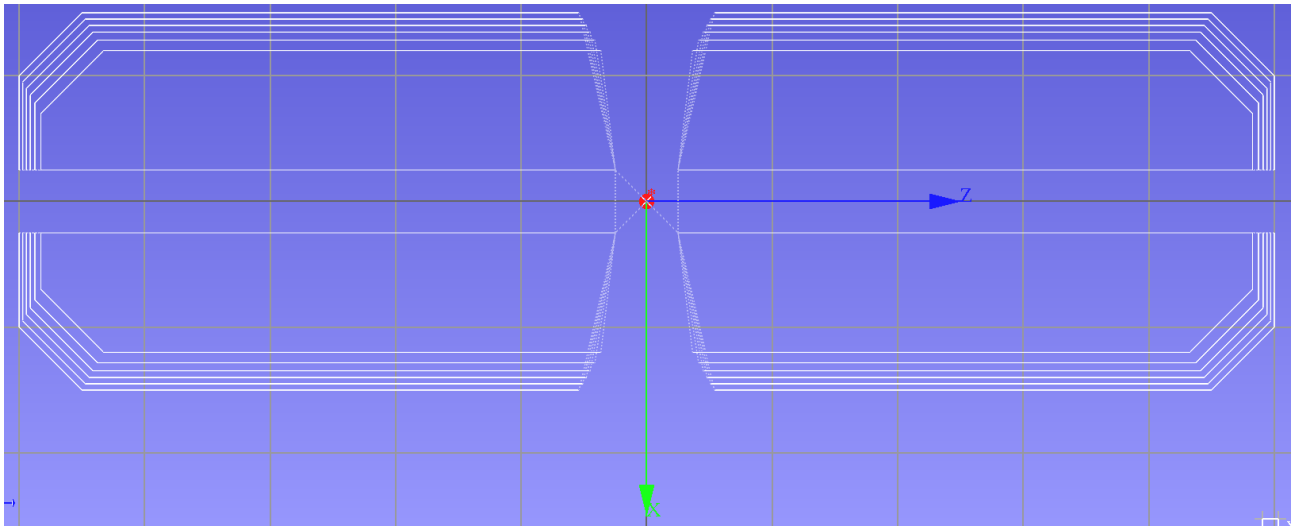


Рис. 2.14.8: Нарезание внутренней и наружной резьбы

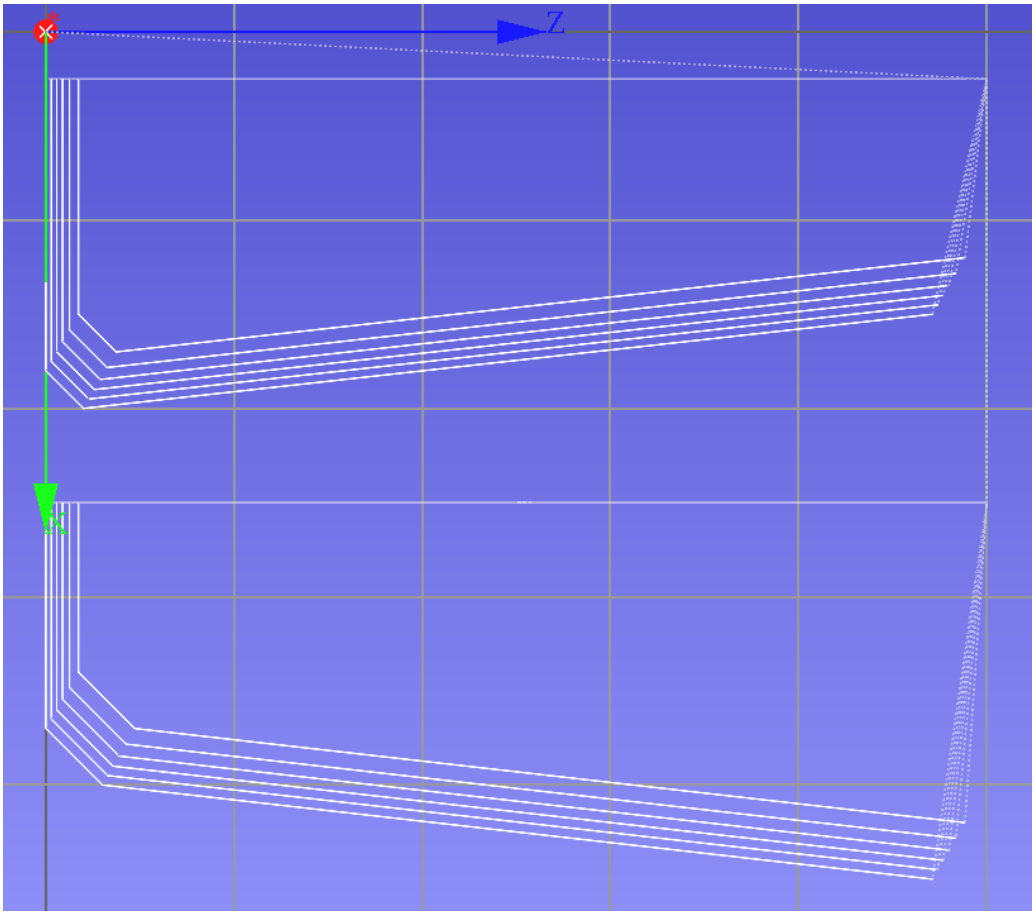


Рис. 2.14.9: Демонстрация нарезания конусной резьбы

2.14.4. Цикл нарезания резьбы с указанием проходов (G78)

Формат (полный):

```
G78 P______ Q__ R__ OA__ ;первый блок команды
G78 X{U}__ Z{W}__ P__ Q__ R__ TT__ F__ C__ ;второй блок команды
;Кадры с заданием координаты X.
G80
```

```
G78 LC__ OD__ TA__ Q__ R__ OA__ ;первый блок команды
G78 X{U}__ Z{W}__ P__ Q__ R__ TT__ F__ C__ ;второй блок команды
;Кадры с заданием координаты X.
G80
```

Цикл **G78** аналогичен циклу G76, но позволяет нарезать резьбу с указанием произвольной глубины каждого прохода. Цикл автоматически вычисляет позицию врезания для глубины, заданной очередным кадром. Кадры внутри цикла можно задавать как через G90, так и через G91. В режиме G91 кадр будет интерпретирован как приращение глубины резьбы по отношение к предыдущему проходу.

Аргументы **первого блока** команды **G78**

LC — число чистовых проходов n после нарезания резьбы, может быть задано параметром N3402.

OD — расстояние r выхода инструмента из резьбы после её нарезания под углом 45 градусов, может быть задано параметром N3401.

TA — угол инструмента a (отвечает за угол входа в резьбу по умолчанию), может быть задан в таблице инструментов для текущего инструмента.

Pnnrraa — целое число, состоящее из 6 цифр (nnrraa), необязательный аргумент (может быть заменён параметрами LC, OD и TA). Каждые две цифры обозначают отдельный параметр:

- nn — аналогичен аргументу **LC**;
- rr — аналогичен аргументу **OD**;
- aa — аналогичен аргументу **TA**.

OA — угол выхода из резьбы. Необязательный параметр (значение по умолчанию: 45 градусов).

Аргументы **второго блока** команды **G78**

X(U) — координата окончания резьбы по декартовой оси X (название оси и аргумента может не совпадать), точка D, мм.

Z(W) — координата окончания резьбы по декартовой оси Z (название оси и аргумента может не совпадать), точка D, мм.

P — глубина резьбы h (радиусное значение), мм.

R — разница ΔR между начальным и конечным радиусами резьбы, мм. Можно указывать отрицательное значение. Используется для нарезания конической резьбы. Необязательный аргумент, значение по умолчанию: $\Delta R = 0$.

TTlcr — целое число из 3 цифр (l, c, и r), задаёт варианты входа резца в резьбу. Необязательный параметр (значение по умолчанию: 001). Возможные значения для lcr:

- 001 — вход в резьбу под углом справа, нагрузка на левую кромку резца (по умолчанию);
- 100 — вход в резьбу под углом слева, нагрузка на правую кромку резца;
- 101 — вход в резьбу под углом по очереди справа и слева, а нагрузка, соответственно, по очереди на левую и правую кромку резца;
- 010 — вход в резьбу по центру с нагрузкой на обе кромки резца;
- 111 — вход в резьбу по центру с нагрузкой по очереди на левую и правую кромки резца.

C — угол начала резьбы (позиция шпинделя относительно 0-метки, с которой необходимо начинать проход резьбы). Необязательный параметр.

F — шаг нарезания резьбы (оборотная подача), мм/об.

Пример нарезания резьбы через G78:

```
G0 X40 Z0 ;координата выхода из очередного прохода цикла
G78 P010060
G78 X30 Z-20 P1 F2
X29.8 ;первый проход
X29.6 ;второй проход
X29.4 ;третий проход
X29.3 ;третий проход
X29.2 ;четвёртый проход
X29.1 ;четвёртый проход
X29.0 ;пятый проход
G80 ;конец цикла
```

2.14.5. Цикл сверления(G81,G82,G83)

G81 — многопроходный цикл сверления с дроблением стружки

Формат (полный):

```
G81 X{U}_ C{H}_ Z{W}_ R_ Q_ D_ K_ M_ F_
```

Аргументы команды **G81**

X(U) — координата отверстия по декартовой оси X (название оси и аргумента может не совпадать).

C(H) — координата отверстия по декартовой оси C (название оси и аргумента может не совпадать).

Z(W) — координата отверстия по декартовой оси Z (название оси и аргумента может не совпадать).

R — координата начала сверления по оси сверления (Z).

Q — шаг сверления по оси Z.

D — отскок по оси Z.

K — количество повторов цикла сверления.

M — M-функция, исполняемая при достижения дна отверстия.

F — подача, используемая при сверлении.

Описание цикла:

1. Движемся по оси X в координату, заданную аргументом X, на быстром ходу.
2. Движемся по оси C в координату, заданную аргументом C, на быстром ходу.
3. Запоминается текущая координата по оси Z.
4. Движемся по оси Z в координату, заданную аргументом R, на быстром ходу.
5. Движемся по оси Z в координату, отличную от текущей на шаг сверления Q, на рабочем ходу.
6. Движемся по оси Z в координату, отличную от текущей на отскок D, на рабочем ходу.

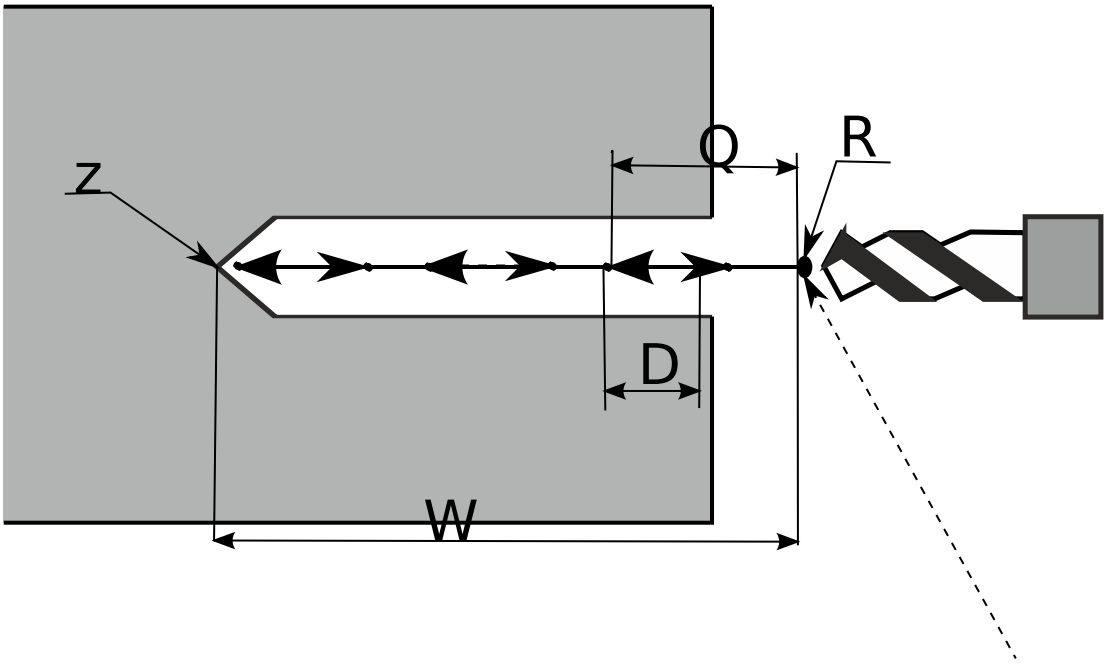


Рис. 2.14.10: Многопроходный цикл сверления с дроблением стружки G81

7. Пункты 5-6 повторяются до тех пор, пока не достигнем координаты по оси Z , заданной аргументом Z.
8. Выполняется М-фунция.
9. Движемся по оси Z в координату, заданную аргументом R, на рабочем ходу.
10. Выходим в начальную координату по оси Z (пункт 3).
11. Если задан аргумент K, то пункты 1-10 повторяются столько раз, сколько задано в аргументе K.

Схематическое обозначение многопроходного цикла сверления с дроблением стружки G81 изображено на рисунке2.14.10

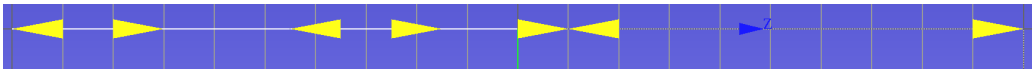


Рис. 2.14.11: Демонстрация цикла G81 с одним отверстием

Пример использования цикла G81 с одним отверстием (рис. 2.14.11)

```
;Начальное положение X0 Z10
G0X0Z10
;Вызываем цикл G81 с параметрами
;Конечная точка сверления Z-10 (параметр Z)
;Начальная точка сверления Z1 (параметр R)
;Шаг сверления 5.5 (параметр Q)
;Шаг возврата 3 (параметр D)
;Скорость сверления 0.7 (параметр F)
G81 Z-10 R1 Q5.5 D3 F0.7
;Отмена цикла G81
G80
G0X5
```

Пример использования цикла G81 с несколькими отверстием (рис. 2.14.12)

```
;Начальное положение X0 Z10
G0X0Z10
;Вызываем цикл G81 с параметрами
;Конечная точка сверления Z-10 (параметр Z)
;Начальная точка сверления Z1 (параметр R)
;Шаг сверления 5.5 (параметр Q)
;Шаг возврата 3 (параметр D)
;Скорость сверления 0.7 (параметр F)
G81 Z-10 R1 Q5.5 D3 F0.7
;Новая координата вызова цикла
G0X5
;Новая координата вызова цикла
G0X7
;Отмена цикла G81
G80
G0X5
```

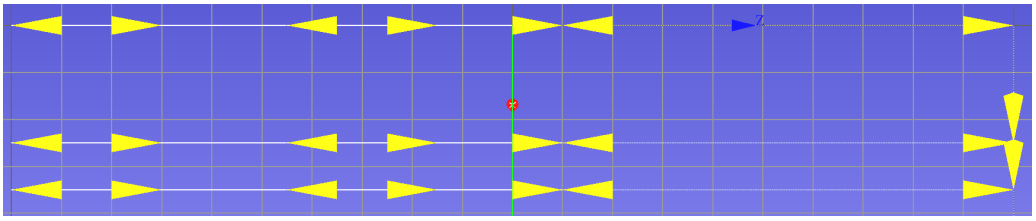


Рис. 2.14.12: Демонстрация цикла G81 с несколькими отверстиями

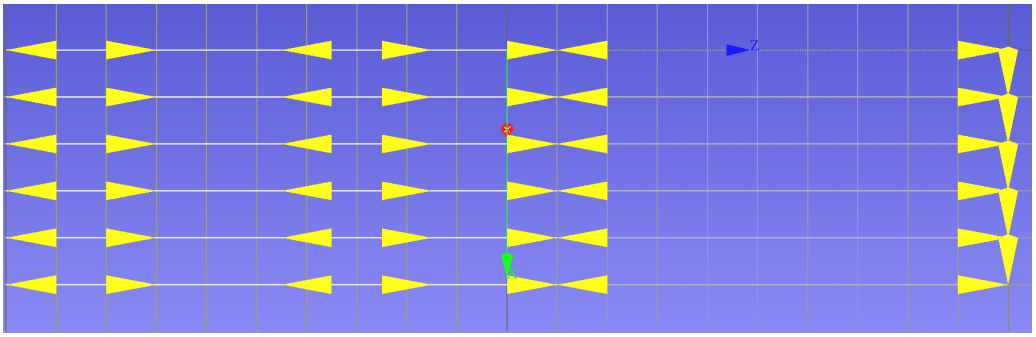


Рис. 2.14.13: Демонстрация цикла G81 с параметром K в режиме G91

Пример использования цикла G81 с параметром K при G91 (рис. 2.14.13):

```
;Начальное положение X0 Z10
G0X0Z10
;Включаем режим G91
G91
;Вызываем цикл G81 с параметрами
;Конечная точка сверления Z-10 (параметр Z)
;Начальная точка сверления Z1 (параметр R)
;Шаг сверления 5.5 (параметр Q)
;Шаг возврата 3 (параметр D)
;Скорость сверления 0.7 (параметр F)
;Количество повторений 5 (параметр K)
G81 Z-20 R-9 Q5.5 D3 K5 F0.7
;Новая координата вызова цикла
G0X2
;Отмена цикла G81
G80
G0X12
```

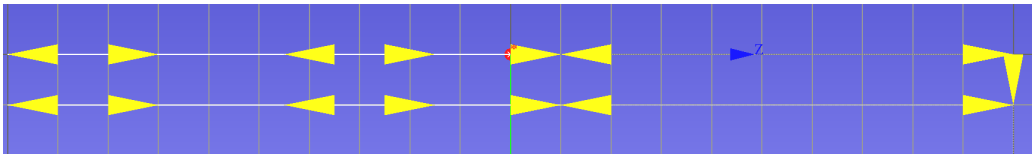


Рис. 2.14.14: Демонстрация цикла G81 с параметром K в режиме G90

Пример использования цикла G81 с параметром K при G90 (рис. 2.14.14):

```
;Начальное положение X0 Z10
G0X0Z10
;Включаем режим G90
G90
;Вызываем цикл G81 с параметрами
;Конечная точка сверления Z-10 (параметр Z)
;Начальная точка сверления Z1 (параметр R)
;Шаг сверления 5.5 (параметр Q)
;Шаг возврата 3 (параметр D)
;Скорость сверления 0.7 (параметр F)
;Количество повторений 5 (параметр K)
G81 Z-10 R1 Q5.5 D3 K5 F0.7
;Новая координата вызова цикла
G0X2
;Отмена цикла G81
G80
G0X12
```

G82— многопроходный цикл сверления с паузой

Формат (полный):

```
G82 Z{W}_ X{U}_ C{H}_ R_ Q_ P_ K_ M_ F_
```

Аргументы команды **G82**

X(U) — координата отверстия по декартовой оси X (название оси и аргумента может не совпадать).

C(H) — координата отверстия по декартовой оси Y (название оси и аргумента может не совпадать).

Z(W) — координата отверстия по декартовой оси Z (название оси и аргумента может не совпадать).

R — координата начала сверления по оси сверления (Z).

Q — шаг сверления по оси Z.

P — пауза в секундах, выдерживаемая в конце шага сверления.

K — количество повторений цикла сверления.

M — М-функция, исполняемая на дне отверстия.

F — подача, используемая при сверлении.

Описание цикла:

1. Движемся по оси X в координату, заданную аргументом X, на быстром ходу.
2. Движемся по оси C в координату, заданную аргументом C, на быстром ходу.
3. Запоминается текущая координата по оси Z.
4. Движемся по оси Z в координату, заданную аргументом R, на быстром ходу.
5. Движемся по оси Z в координату, отличную от текущей на шаг сверления, заданный аргументом Q.
6. Выдерживаем паузу заданную аргументом P.
7. Повторяем пункты 5-6 до тех пор, пока не достигнем координаты по оси Z , заданной аргументом Z.
8. Выполняется М-фунция.
9. Выключаем шпиндель и движемся по оси Z в координату, заданную аргументом R.
10. Выходим в начальные координаты по оси Z (пункт 3).
11. Если задан аргумент K, то пункты 1-10 повторяются столько раз, сколько задано аргументом K.

Схематическое обозначение многопроходного цикла сверления с паузой G82 изображено на рисунке 2.14.15

G83— многопроходный цикл сверления с полным выводом сверла из отверстия

Формат (полный):

```
G83 Z{W}_ X{U}_ C{H}_ R_ Q_ D_ P_ K_ M_ F_
```

Аргументы команды **G83**

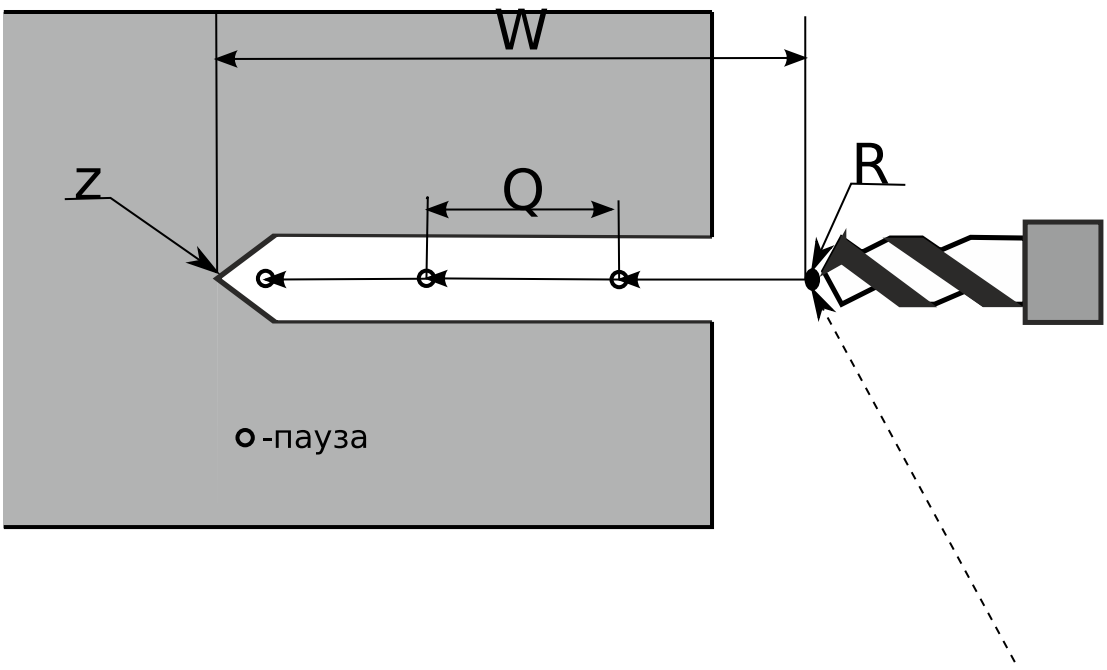


Рис. 2.14.15: Многопроходный цикл сверления с паузой G82

X(U) — координата отверстия по декартовой оси X (название оси и аргумента может не совпадать).

C(H) — координата отверстия по декартовой оси C (название оси и аргумента может не совпадать).

Z(W) — координата отверстия по декартовой оси Z (название оси и аргумента может не совпадать).

R — координата начала сверления по оси сверления (Z).

Q — шаг сверления по оси Z.

D — расстояние до начала следующего шага.

P — пауза в секундах, выдерживаемая в конце шага сверления.

K — количество повторов цикла сверления.

M — M-функция, исполняемая на дне отверстия.

F — подача, используемая при сверлении.

Описание цикла:

1. Движемся по оси X в координату, заданную аргументом X, на быстром ходу.
2. Движемся по оси C в координату, заданную аргументом C, на быстром ходу.

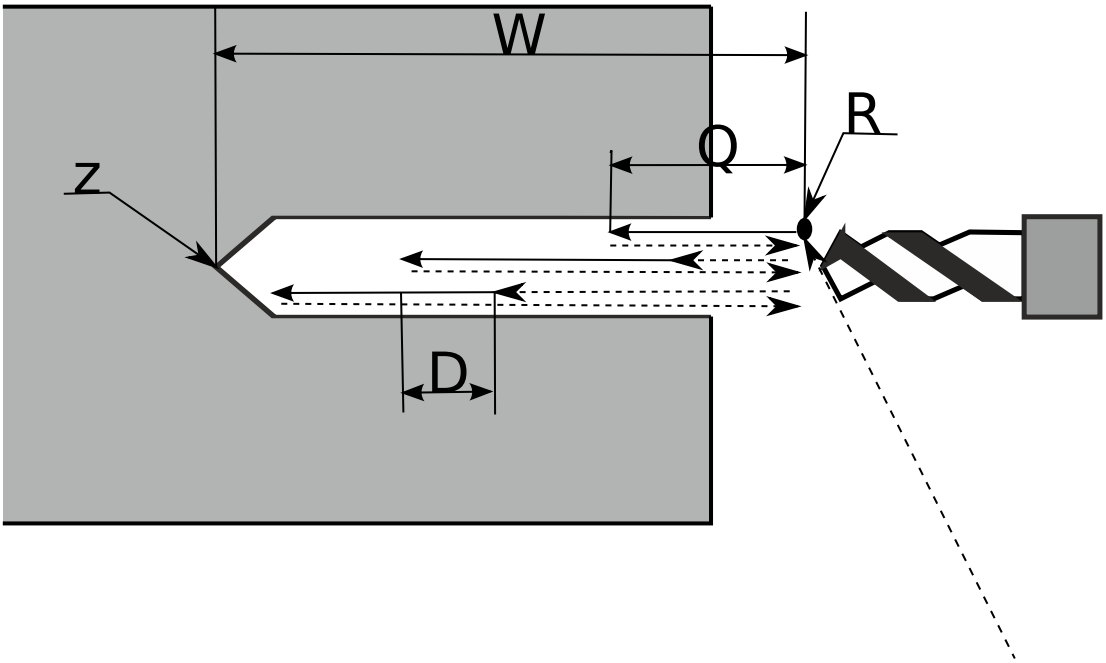


Рис. 2.14.16: Многопроходный цикл сверления с полным выводом сверла из отверстия G83

3. Запоминается текущая координата по оси Z (для пункта 11).
4. Движемся по оси Z в координату, заданную аргументом R, на быстром ходу.
5. Движемся по оси Z в координату, отличную от текущей на шаг сверления, заданный аргументом Q.
6. Движемся по оси Z в координату, заданную аргументом R .
7. Движемся по оси Z в координату, отличную от конца шага на аргумент D.
8. Повторяем пункты 4-7 до тех пор, пока не достигнем координаты по оси Z, заданной аргументом Z.
9. Выполняется M-фунция.
10. Движемся по оси Z в координату, заданную аргументом R.
11. Выходим в начальные координаты по оси Z (пункт 3).
12. Если задан аргумент K, то пункты 1-11 повторяются столько раз, сколько задано аргументом K.

Схематическое обозначение многопроходного цикла сверления с полным выводом сверла из отверстия G83 изображено на рисунке2.14.16

2.14.6. Цикл нарезания резьбы метчиком G84

Формат (полный):

```
G84 Z{W}_ X{U}_ C{H}_ R_ Q_ K_ MR_ MC_ F_
```

Аргументы команды **G84**

X(U) — координата резьбы по декартовой оси X (название оси и аргумента может не совпадать).

C(H) — координата резьбы по декартовой оси C (название оси и аргумента может не совпадать).

Z(W) — координата резьбы по декартовой оси Z (название оси и аргумента может не совпадать).

R — координата начала резьбы по оси Z.

Q — приращение глубины резьбы на один проход по оси Z.

K — количество повторов цикла нарезания резьбы.

MR — M-функция реверса шпинделя.

MC — M-функция отмены реверса шпинделя.

F — подача, используемая при нарезании резьбы.

Описание цикла:

1. Движемся по оси X в координату, заданную аргументом X, на быстром ходу.
2. Движемся по оси C в координату, заданную аргументом C, на быстром ходу.
3. Запоминается текущая координата по оси Z (для пункта 11).
4. Движемся по оси Z в координату, заданную аргументом R, на быстром ходу.
5. Поиск ноль-метки.
6. Если аргумент Q не задан, то нарезание резьбы пройдет за один шаг.
7. Опускаемся по оси Z в координату текущей глубины резьбы (вычисленной для текущего прохода исходя из Q), на рабочем ходу.
8. Поднимаемся по оси Z на величину Q на рабочем ходу.

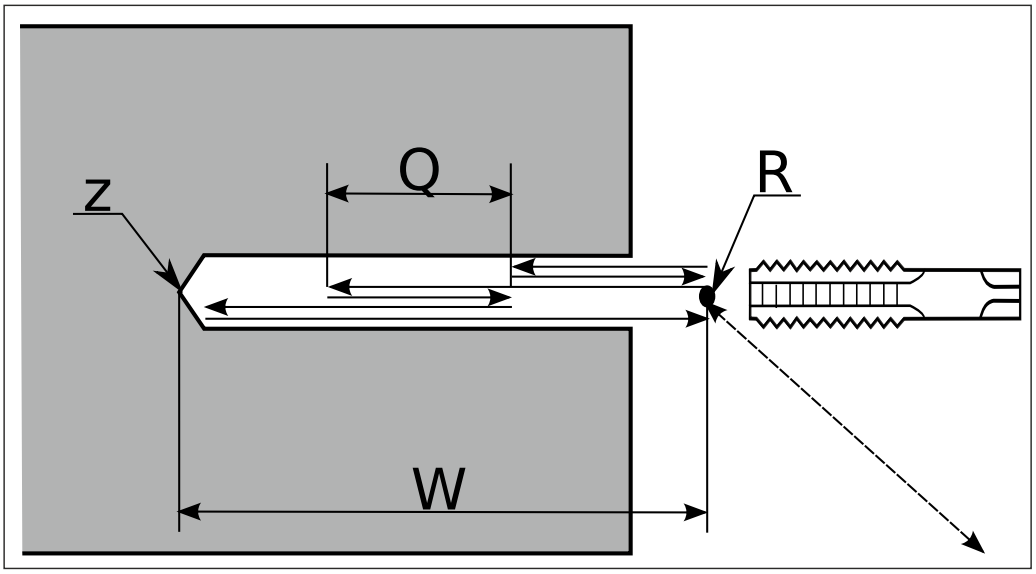


Рис. 2.14.17: Цикл нарезания резьбы метчиком (с плавающей головкой), G84

9. Пункты 7-8 повторяются до тех пор, пока по оси Z не достигнем координаты, заданной аргументом Z.
10. Движемся по оси Z в координату, заданную аргументом R, на рабочем ходу.
11. Выходим в начальную координату по оси Z (из пункта 3).
12. Если задан аргумент K, то пункты 1-10 повторяются K раз.

Пример нарезания резьбы метчиком (с плавающей головкой) с помощью цикла G84 (рис. 2.14.17):

```
;Запуск шпинделя в режиме M3
M3
;Выходим в начальную точку запуска цикла G84
G1 Z10 F1
;Вызываем цикл G84 с параметрами
;Конечная точка нарезания резьбы Z-10 (параметр Z)
;Начальная точка нарезания резьбы Z1 (параметр R)
;Скорость нарезания резьбы 0.7 (параметр F)
;M-функция реверса M34 (параметр MR)
;M-функция отмены реверса M35 (параметр MC)
;Приращение глубины резьбы на один проход 4 (параметр Q)
G84 Z-10 R1 F0.7 MR34 MC35 Q4
;Отмена цикла G84
G80
```

Примечание:

1. Для корректной работы цикла нужна реализация M-функции реверса шпинделя (раздел Реверс шпинделя в документации по PLC).
2. M-функция реверса шпинделя должна быть прописана в параметре 3100.

2.15 Сверлильные циклы

2.15.1. Многопроходный цикл сверления G81

Формат (полный):

```
G81 X{U}_ Y{V}_ Z{W}_ R_ Q_ D_ K_ M_ F_
```

Аргументы команды **G81**

X(U) — координата отверстия по декартовой оси X (название оси и аргумента может не совпадать).

Y(V) — координата отверстия по декартовой оси Y (название оси и аргумента может не совпадать).

Z(W) — координата отверстия по декартовой оси Z (название оси и аргумента может не совпадать).

R — координата начала сверления по оси сверления (Z).

Q — шаг сверления по оси Z.

D — отскок по оси Z.

K — количество повторов цикла сверления.

M — M-функция, исполняемая при достижения дна отверстия.

F — подача, используемая при сверлении.

Описание цикла:

1. Движемся по оси X в координату, заданную аргументом X, на быстром ходу.
2. Движемся по оси Y в координату, заданную аргументом Y, на быстром ходу.
3. Запоминается текущая координата по оси Z.

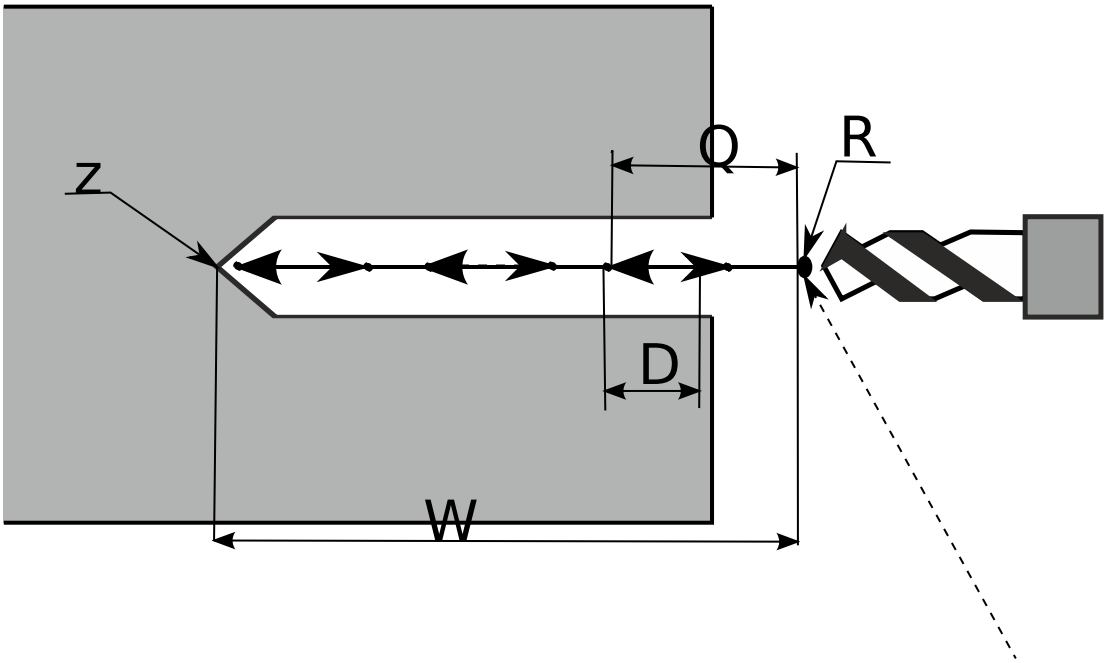


Рис. 2.15.1: Многопроходный цикл сверления с дроблением стружки G81

4. Движемся по оси Z в координату, заданную аргументом R, на быстром ходу.
5. Движемся по оси Z в координату, отличную от текущей на шаг сверления Q, на рабочем ходу.
6. Движемся по оси Z в координату, отличную от текущей на отскок D, на рабочем ходу.
7. Пункты 5-6 повторяются до тех пор, пока не достигнем координаты по оси Z, заданной аргументом Z.
8. Выполняется M-фунция.
9. Движемся по оси Z в координату, заданную аргументом R, на рабочем ходу.
10. Выходим в начальную координату по оси Z (пункт 3).
11. Если задан аргумент K, то пункты 1-10 повторяются столько раз, сколько задано в аргументе K.

Схематическое обозначение многопроходного цикла сверления с дроблением стружки G81 изображено на рисунке2.15.1

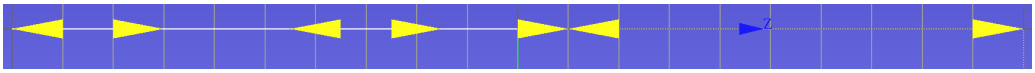


Рис. 2.15.2: Демонстрация цикла G81 с одним отверстием

Пример использования цикла G81 с одним отверстием (рис. 2.15.2):

```
;Начальное положение X0 Z10
G0X0Z10
;Вызываем цикл G81 с параметрами
;Конечная точка сверления Z-10 (параметр Z)
;Начальная точка сверления Z1 (параметр R)
;Шаг сверления 5.5 (параметр Q)
;Шаг возврата 3 (параметр D)
;Скорость сверления 0.7 (параметр F)
G81 Z-10 R1 Q5.5 D3 F0.7
;Отмена цикла G81
G80
G0X5
```

Пример использования цикла G81 с несколькими отверстием (рис. 2.15.3):

```
;Начальное положение X0 Z10
G0X0Z10
;Вызываем цикл G81 с параметрами
;Конечная точка сверления Z-10 (параметр Z)
;Начальная точка сверления Z1 (параметр R)
;Шаг сверления 5.5 (параметр Q)
;Шаг возврата 3 (параметр D)
;Скорость сверления 0.7 (параметр F)
G81 Z-10 R1 Q5.5 D3 F0.7
;Новая координата вызова цикла
G0X5
;Новая координата вызова цикла
G0X7
;Отмена цикла G81
G80
G0X5
```

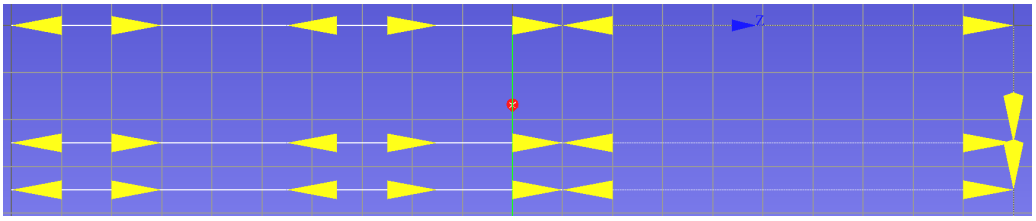


Рис. 2.15.3: Демонстрация цикла G81 с несколькими отверстиями

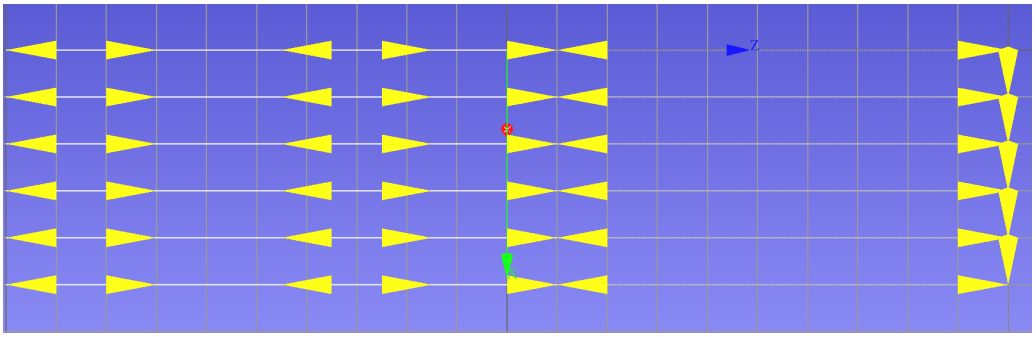


Рис. 2.15.4: Демонстрация цикла G81 с параметром K в режиме G91

Пример использования цикла G81 с параметром K при G91 (рис. 2.15.4):

```
;Начальное положение X0 Z10
G0X0Z10
;Включаем режим G91
G91
;Вызываем цикл G81 с параметрами
;Конечная точка сверления Z-10 (параметр Z)
;Начальная точка сверления Z1 (параметр R)
;Шаг сверления 5.5 (параметр Q)
;Шаг возврата 3 (параметр D)
;Скорость сверления 0.7 (параметр F)
;Количество повторений 5 (параметр K)
G81 Z-20 R-9 Q5.5 D3 K5 F0.7
;Новая координата вызова цикла
G0X2
;Отмена цикла G81
G80
G0X12
```

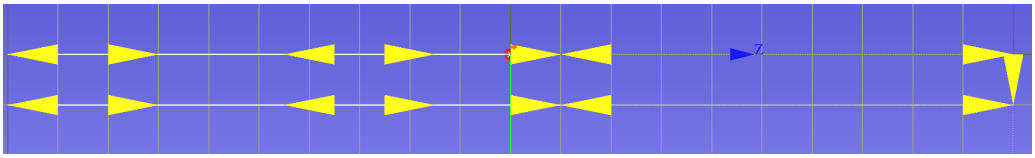


Рис. 2.15.5: Демонстрация цикла G81 с параметром K в режиме G90

Пример использования цикла G81 с параметром K при G90 (рис. 2.15.5):

```
;Начальное положение X0 Z10
G0X0Z10
;Включаем режим G90
G90
;Вызываем цикл G81 с параметрами
;Конечная точка сверления Z-10 (параметр Z)
;Начальная точка сверления Z1 (параметр R)
;Шаг сверления 5.5 (параметр Q)
;Шаг возврата 3 (параметр D)
;Скорость сверления 0.7 (параметр F)
;Количество повторений 5 (параметр K)
G81 Z-10 R1 Q5.5 D3 K5 F0.7
;Новая координата вызова цикла
G0X2
;Отмена цикла G81
G80
G0X12
```

2.15.2. Многопроходный цикл сверления с паузой G82

Формат (полный):

```
G82 Z{W}_ X{U}_ Y{V}_ R_ Q_ P_ K_ M_ F_
```

Аргументы команды **G82**

X(U) — координата отверстия по декартовой оси X (название оси и аргумента может не совпадать).

Y(V) — координата отверстия по декартовой оси Y (название оси и аргумента может не совпадать).

Z(W) — координата отверстия по декартовой оси Z (название оси и аргумента может не совпадать).

R — координата начала сверления по оси сверления (Z).

Q — шаг сверления по оси Z.

P — пауза в секундах, выдерживаемая в конце шага сверления.

K — количество повторений цикла сверления.

M — M-функция, исполняемая на дне отверстия.

F — подача, используемая при сверлении.

Описание цикла:

1. Движемся по оси X в координату, заданную аргументом X, на быстром ходу.
2. Движемся по оси Y в координату, заданную аргументом Y, на быстром ходу.
3. Движемся по оси Z в координату, заданную аргументом R, на быстром ходу.
4. Происходит включение шпинделя.
5. Движемся по оси Z в координату, отличную от текущей на шаг сверления, заданный аргументом Q.
6. Выдерживаем паузу, заданную аргументом P.
7. Повторяем пункты 5-6 до тех пор, пока не достигнем по оси Z координаты, заданной аргументом Z.
8. Выполняется M-фунция.
9. Выключаем шпиндель и движемся по оси Z в координату, заданную аргументом R.
10. Выходим в начальную координату по оси Z.
11. Если задан аргумент K, то пункты 1-10 повторяются столько раз, сколько задано аргументом K.

Схематическое обозначение многопроходного цикла сверления G82 изображено на рисунке 2.15.6.

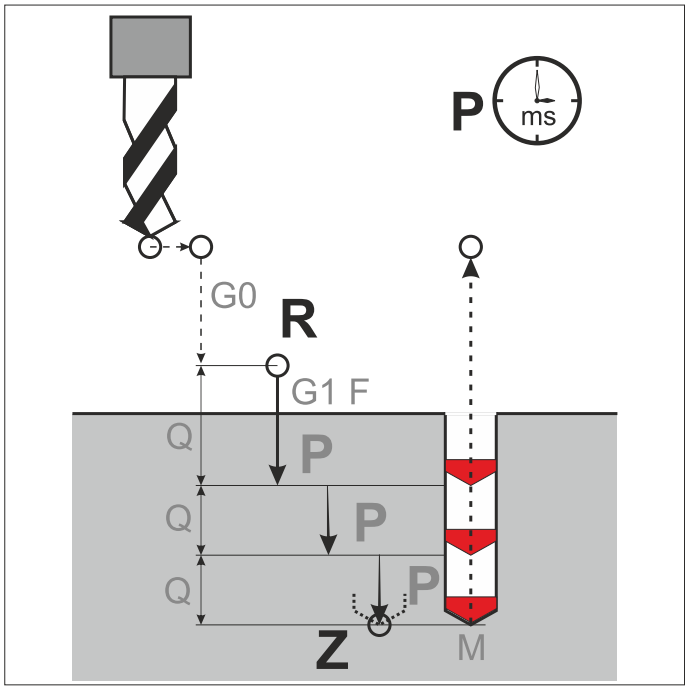


Рис. 2.15.6: Многопроходный цикл сверления с паузой G82

2.15.3. Многопроходный цикл сверления с полным выводом сверла из отверстия G83

Формат (полный):

```
G83 Z{W}_ X{U}_ Y{V}_ R_ Q_ D_ P_ K_ M_ F_
```

Аргументы команды **G83**

X(U) — координата отверстия по декартовой оси X (название оси и аргумента может не совпадать).

Y(V) — координата отверстия по декартовой оси Y (название оси и аргумента может не совпадать).

Z(W) — координата отверстия по декартовой оси Z (название оси и аргумента может не совпадать).

R — координата начала сверления по оси сверления (Z).

Q — шаг сверления по оси Z.

D — расстояние до начала следующего шага.

P — пауза в секундах, выдерживаемая в конце шага сверления.

K — количество повторов цикла сверления.

M — M-функция, исполняемая на дне отверстия.

F — подача, используемая при сверлении.

Описание цикла:

1. Движемся по оси X в координату, заданную аргументом X, на быстром ходу.
2. Движемся по оси Y в координату, заданную аргументом Y, на быстром ходу.
3. Движемся по оси Z в координату, заданную аргументом R, на быстром ходу.
4. Происходит включение шпинделя.
5. Движемся по оси Z в координату, отличную от текущей на шаг сверления, заданный аргументом Q.
6. Движемся по оси Z в координату, заданную аргумент R .
7. Движемся по оси Z в координату, отличную от конца шага на аргумент D.
8. Повторяем пункты 5-7 до тех пор, пока не достигнем по оси Z в координату, заданную аргументом Z.
9. Выполняется M-фунция.
10. Выключаем шпиндель и движемся по оси Z в координату, заданную параметром R.
11. Выходим в начальные координаты по оси Z.
12. Если задан аргумент K, то пункты 1-11 повторяются столько раз, сколько задано аргументом K.

Схематическое обозначение многопроходного цикла сверления G83 изображено на рисунке 2.15.7.

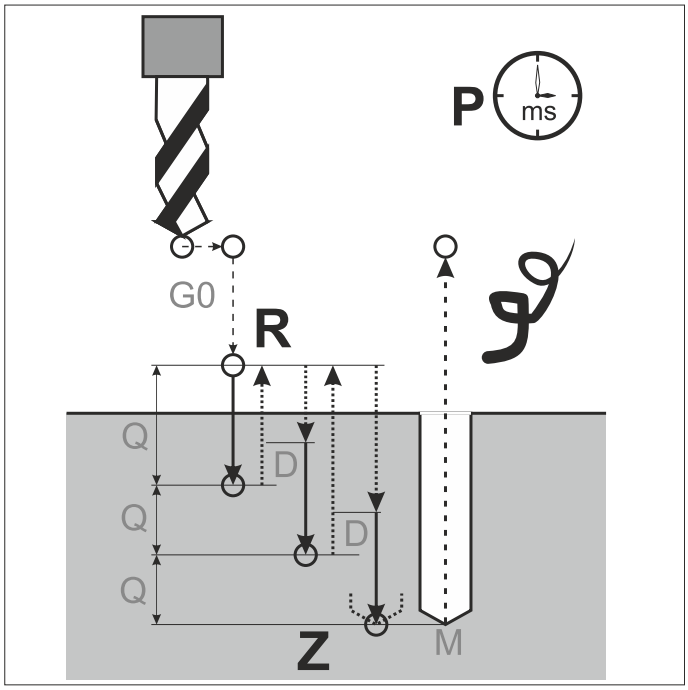


Рис. 2.15.7: Многопроходный цикл сверления с полным выводом сверла из отверстия G83

2.15.4. Подпрограмма сверления отверстий по окружности G65 P9030

Формат (полный):

```
G65 P9030 A_ D_ P_ Z_ R_ Q_ F_
```

Аргументы команды **G65 P9030**

A — угол смещения начальной точки по окружности.

D — диаметр окружности.

P — количество отверстий по окружности.

Z — глубина каждого отверстия.

R — координата начала сверления по оси Z.

Q — шаг сверления по оси Z.

F — подача, используемая при сверлении.

Описание цикла:

1. Сдвигаемся на угол, заданный аргументом A, от начальной точки.
2. Окружность разбивается на количество отверстий, заданных аргументом P, вычисляем угол смещения.

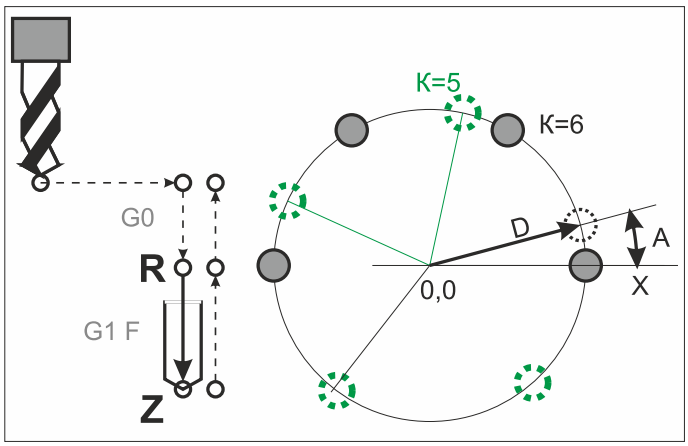


Рис. 2.15.8: Подпрограмма сверления отверстий по окружности G65 P9030

3. Движемся по окружности на угол смещения, вычисленный в пункте 2.
4. Сверлим отверстие глубиной, заданной аргументом Z.
5. Повторяем пункты 3-4 по количеству отверстий P.

Схематическое обозначение подпрограммы сверления отверстий по окружности G65 P9030 изображено на рисунке 2.15.8.

2.15.5. Цикл расточки отверстия фрезой G181

Формат (полный):

```
G181 Z{W}_ Y{V}_ X{U}_ R_ D_ ;первый блок команды
G181 D_ Q_ P_ F_ M_ ;второй блок команды
```

Аргументы **первого блока** команды **G181**

X(U) — координата отверстия по декартовой оси X (название оси и аргумента может не совпадать).

Y(V) — координата отверстия по декартовой оси Y (название оси и аргумента может не совпадать).

Z(W) — координата отверстия по декартовой оси Z (название оси и аргумента может не совпадать).

R — координата начала сверления по оси сверления (Z).

D — диаметр отверстия.

Аргументы **второго блока** команды **G181**

D — начальный диаметр отверстия (необязательный параметр).

Q — шаг сверления по оси Z.

P — шаг смещения по оси X.

F — подача, используемая при расточке.

M — M-команда, исполняемая на дне отверстия.

Описание цикла:

1. Движемся по оси X в координату, заданную аргументом X, на быстром ходу.
2. Движемся по оси Y в координату, заданную аргументом Y, на быстром ходу.
3. Движемся по оси Z в координату, заданную аргументом R, на быстром ходу.
4. Движемся по оси X на D/2, если задан аргумент D во втором блоке цикла.
5. Движемся по спиральной траектории на расстояние, заданное аргументом Q.
6. Возвращаемся по оси Z в координату, заданную аргументом R.
7. Смещаемся на оси X на расстояние, заданное аргументом P.
8. Пункты 5-7 повторяются, пока не будет достигнут диаметр отверстия, заданный аргументом D в первом блоке программы.

Схематическое обозначение цикла расточки отверстия фрезой G181 изображено на рисунке 2.15.9.

G17

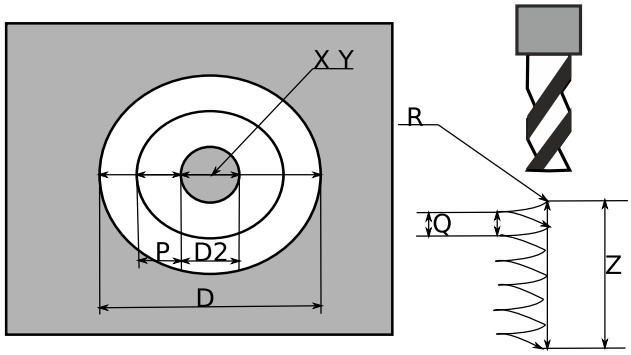


Рис. 2.15.9: Цикл расточки отверстия фрезой G181

2.16 Расточные циклы

2.16.1. Цикл расточки прямоугольного кармана G301

Цикл срезает один или более слоёв с заготовки, входя в очередной слой под заданным углом зигзагом в плоскости обработки. Затем делает чистовой проход по всему контуру, если был указан припуск на чистовой проход.
Формат (полный):

```
G301 [X] [Y] [R__] Z__ [P__] W__ H__ D__ [Q__] [A__] [E__] [F__]
```

Аргументы

X — координата центра кармана по декартовой оси X. Если не указана, берётся текущая координата.

Y — координата центра кармана по декартовой оси Y. Если не указана, берётся текущая координата.

R — координата начала расточки по декартовой оси Z. Если не указана, берётся текущая координата.

Z — координата конца расточки по декартовой оси Z.

P — глубина срезаемого слоя.

W — ширина прямоугольного кармана по декартовой оси X.

H — высота прямоугольного кармана по декартовой оси Y.

D — диаметр фрезы.

Q — ширина срезаемого слоя (после осуществления врезания). Должна быть меньше либо равна диаметру D фрезы.

A — угол входа в заготовку, по умолчанию: 30 градусов.

F — рабочая подача врезания и расточки.

E — припуск, оставляемый на чистовой проход по контуру кармана.

Алгоритм работы цикла

1. Инструмент на ускоренной подаче перемещается в позицию начала зигзага с координатой по Z равной R для первого слоя либо с координатой Z предыдущего слоя.
2. Зигзагом инструмент врезается в заготовку до тех пор, пока не дойдёт до координаты очередной глубины по Z.
3. Инструмент кромкой срезает слой шириной Q или D по прямоугольной спирали из центра к краям кармана в рамках текущего слоя.
4. Пока не достигнули координаты Z, повторяем процесс с пункта 1.
5. Инструмент режущей кромкой срезает ширину E чистового прохода, если она задана.
6. Выход в позицию Z, в которой инструмент находился перед циклом.

Схематическое обозначение цикла расточки прямоугольного кармана G301 изображено на рисунке 2.16.1.

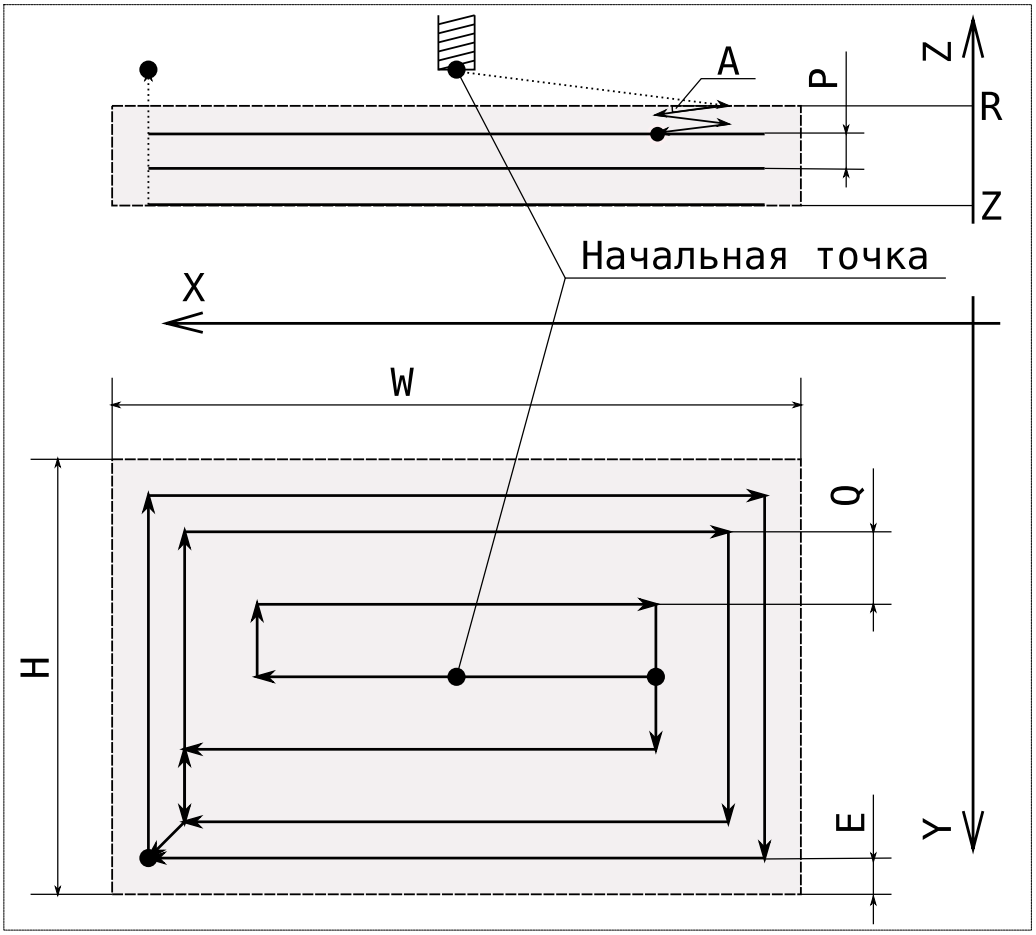


Рис. 2.16.1: Цикл расточки прямоугольного кармана G301

3 Макропрограммирование

3.1 Общие сведения

3.1.1. Типы данных

Вещественное число (FLOAT)

Размер: 8 байт.

Вещественное число с плавающей точкой двойной точности. Диапазон значений: $\pm 2.2250738585072014 \cdot 10^{-308} \ldots \pm 1.7976931348623157 \cdot 10^{-308}$.

Для создания переменной вещественного типа необходимо в качестве значения переменной указать десятичное число.

Вещественные числа позволяют получать особые значения, например, число 0 может иметь знак ($+0.0$ и -0.0). При делении числа на 0 можно получить специальное значение «бесконечность» ($\pm\infty$). А при операции $0.0/0.0$ в качестве результата будет получено специальное значение NaN (Not a Number), которое будет возвращать результат «ЛОЖЬ» в любых операциях сравнения.

Пример. Создание переменной типа FLOAT

```
1 #X = 0.1
2 PRINT #X ;Выведет на экран вещественное число число
```

Целое знаковое число (INT)

Размер: 8 байт.

Целое число со знаком. Диапазон значений:
$-9223372036854775808 \ldots +9223372036854775807$.

Для создания переменной целого типа необходимо в качестве значения переменной указать целое число.

Пример. Создание переменной типа INT

```
#X = 2
PRINT #X ;Выведет на экран целое число
```

Беззнаковый байт (BYTE)

Размер: 1 байт.

Целое число. Диапазон значений: $+0 \ldots +255$.

Стандартных средств для создания переменной данного типа нет. Однако, некоторые встроенные в УЧПУ переменные имеют этот тип данных.

Пример. Выход за пределы диапазона значений

```
#U10 = 255
#U10 = #U10 + 1 ;Из-за переполнения значение переменной установится в 0.
#U10 = 255 + 3 ;Из-за переполнения значение переменной установится в 2.
```

Строковой тип (STRING)

Размер: динамический.

Тип данных, указывающий на строковое значение (текст).

Для создания переменной строкового типа необходимо в качестве значения переменной указать текст в двойных кавычках.

Пример. Создание переменной типа STRING

```
#STR = "Hello World!"
PRINT #STR ;Выведет на экран текстовое сообщение
```

Массивы

Макроязык позволяет создавать динамические массивы переменных, к которым можно обращаться по индексу, указанному в квадратных скобках. Сам по себе массив на самом деле является набором обычных переменных, не обязательно идущих друг за другом в оперативной памяти. Но к элементам массива можно обращаться по индексу, в качестве которого можно использовать переменные и выражения (что облегчает программирование некоторых специфичных задач).

Вследствие особенностей работы массивов, следует учитывать, что, к примеру, переменные #A и #A[1] являются совершенно разными переменными, никак друг с другом не связанными. То есть может существовать отдельная переменная с таким же названием, как и название массива.

Пример. Создание массива

```
#ARR[0] = 0
#ARR[2] = 11
#ARR[5] = 22
PRINT #ARR[2] ;выведет на экран значение 1
PRINT #ARR[3] ;выведет на экран значение 0, т.к. 3-го элемента массива не
   существует
PRINT [ISSET #ARR[0]] ;выведет на экран значение 1, т.к. 0-й элемент
   массива существует
PRINT [ISSET #ARR[1]] ;выведет на экран значение 0, т.к. 1-го элемента
   массива не существует
```

Специальные типы данных

Помимо трёх перечисленных существуют специальные встроенные типы данных: указатели на системные переменные, указатели на вещественные числа одинарной точности и т. п. Многие служебные переменные принадлежат к специальным типам данных.

Оператор присвоения значения

Переменным можно присваивать значения вышеперечисленных типов. Для присвоения значения используется оператор присвоения значения «=». Помимо оператора присвоения существуют

- оператор прибавления значения («+=»),
- оператор вычитания значения («-=»),
- оператор умножения на значение («*=»),
- оператор деления на значение («/=»).

Формат:

```
#VAR = value ;занести в переменную #VAR значение ”value”
#VAR += value ;занести в переменную #VAR значение #VAR + ”value”
#VAR –= value ;занести в переменную #VAR значение #VAR - ”value”
#VAR *= value ;занести в переменную #VAR значение #VAR * ”value”
#VAR /= value ;занести в переменную #VAR значение #VAR / ”value”
```

Приведение типов данных

Приведение типов работает автоматически при использовании арифметических выражений и оператора присваивания.

При операциях между типом FLOAT и INT целое число приводится к вещественному типу и результат получается типа FLOAT.

Пример. Приведение типа INT к FLOAT

```
#X = 2
#Y = 2 * 1.0 ;Переменная #Y стала типа FLOAT, значение - 2.0
```

Тип STRING может быть приведён к типу FLOAT при попытке выполнения над ним каких-либо математических операций.

Пример. Приведение типа STRING к FLOAT

```
#X = "2"
#Y = #X * 1 ;Переменная #Y стала типа FLOAT, значение - 2.0
```

Тип STRING может быть приведён к типу INT, если аргумент типа STRING попытаться передать в фукцию, которая принимает только тип INT.

Пример. Приведение типа STRING к INT

```
#TOOL_NUM = "1.6"
T#TOOL_NUM ;равносильно вызову T1
```

3.1.2. Область видимости переменных

Все переменные, создаваемые внутри программы, являются локальными переменными и хранятся в стеке, специально отведённом для конкретной программы/подпрограммы. Как только программа/подпрограмма завершает своё исполнение, её стек переменных автоматически очищается. Т. е. все переменные, объявленные внутри файла, видны только в нём. Как только файл завершает своё исполнение, его переменные уничтожаются.

Циклические подпрограммы сохраняют значение своих переменных до окончания всех итераций цикла.

3.1.3. Получение координат по осям

В ходе исполнения программы от кадра к кадру меняются координаты инструмента. Интерпретатор за основу расчётов начальной и конечной точки траектории берёт не реальные координаты, а вычисленные. Поэтому, если при исполнении программы произошло смещение инструмента с расчётной траектории (например, в M- или T-команде), интерпретатор будет продолжать кадры без учёта смещения траектории.

Если возникает необходимость корректировать траекторию в соответствии с новыми координатами осей в заданной точке программы, то можно получить от интерполятора текущие координаты по любой из осей, обратившись к ней по имени. При этом программа приостановит свою работу до тех пор, пока все предыдущие кадры не будут исполнены. И только после этого будет возвращена текущая рабочая координата по заданной оси.

Пример. Получение координаты оси X

```
1 G0 X10
2 T3 ;произошло перемещение в координату X=30
3 PRINT X ;выведет число 30 после того, как отработает команда T3
```

Если же необходимо получить расчётную рабочую координату в заданной точке программы (предполагая, что траектория движения измениться не может), то можно воспользоваться служебным массивом #_AXIS_POS[axis_num], указав вместо «axis_num» порядковый номер оси (нумерация от 1-цы).

Для получения текущей абсолютной координаты — массив #_AXIS_ABS_POS[axis_num].

3.2 Основные операторы и переменные

3.2.1. Арифметические операторы

Арифметические операции включают в себя стандартный набор операций над целыми и вещественными числами. Все арифметические операции можно группировать при помощи квадратных скобок. Без группировки порядок вычислений определяется приоритетом операций. В таблице 3.2.1 приводится список арифметических операций, где a и b — аргументы арифметической операции (константы, переменные, операторы, оси либо выражения).

Помимо стандартных математических операций существует набор функций для вычисления различных математических значений. В таблице 3.2.2 представлены примеры операторов.

Далее приводится более подробное описание каждой из математический функций.

SQRT value

Возвращает квадратный корень из аргумента «value».

Результат: значение типа FLOAT.

CBRT value

Возвращает кубический корень из аргумента «value».
Результат: значение типа FLOAT.

ROUND value

Возвращает значение «value», округлённое до ближайшего целого.
Результат: значение типа INT.

FIX value

Возвращает значение «value», округлённое до ближайшего целого, меньшего либо равного заданному.
Результат: значение типа INT.

FUP value

Возвращает значение «value», округлённое до ближайшего целого, большего либо равного заданному.
Результат: значение типа INT.

TRUNC value

Возвращает целую часть аргумента «value».
Результат: значение типа INT.

MODF value

Возвращает дробную часть аргумента «value».
Результат: значение типа FLOAT.

ABS value

Возвращает значение аргумента «value», взятое по модулю.
Если «value» — целое число типа INT, то результат — целое число, в противном случае будет возвращено вещественное число типа FLOAT.

RND value

Возвращает случайное число в диапазоне от 0 до значения аргумента «value».

Если «value» — целое число типа INT, то результат — целое число, в противном случае будет возвращено вещественное число типа FLOAT.

LN value

Возвращает натуральный логарифм аргумента «value».
Результат: значение типа FLOAT.

LOG10 value

Возвращает логарифм по основанию 10 аргумента «value».
Результат: значение типа FLOAT.

EXP value

Возвращает экспоненту аргумента «value».
Результат: значение типа FLOAT.

BCD value

Возвращает аргумент «value», переведённый в формат двоичного числа BCD.
Результат: значение типа INT.

BIN value

Возвращает аргумент «value», переведённый из формата двоичного числа BCD в формат обычного десятичного числа.
Результат: значение типа INT.

SIN angle

Единица измерения аргумента «angle»: градусы.
Возвращает синус угла «angle» в диапазоне [-1.0; +1.0].
Результат: значение типа FLOAT.
На рисунке 3.2.1 показана зависимость возвращаемого результата от аргумента «angle».

COS angle

Единица измерения аргумента «angle»: градусы.
Возвращает косинус угла «angle» в диапазоне [-1.0; +1.0].

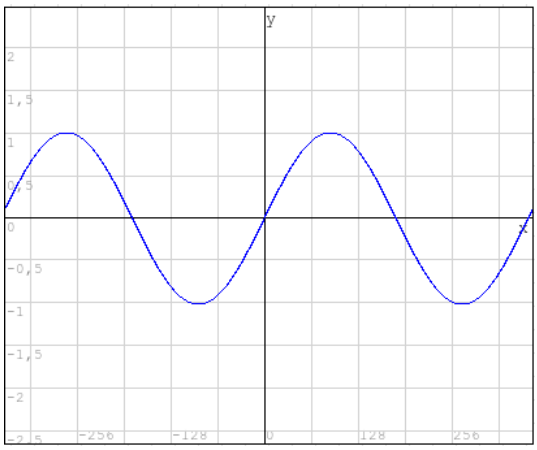


Рис. 3.2.1: График оператора sin

Результат: значение типа FLOAT.

Пример. Интерполяция по спирали средствами макроязыка

```
;Входные данные:
#XC = 10
#YC = 10
#A = 0
#R = 10
;Подготовка:
G0 Z10
#X = #XC + #R * COS #A
#Y = #YC + #R * SIN #A
G0 X#X Y#Y
#Z = 0
G0 Z#Z
;Исполнение:
WHILE [ #R >= 0 ] DO
    #X = #XC + #R * COS #A
    #Y = #YC + #R * SIN #A
    G1 X#X Y#Y Z#Z F500
    #A = #A + 1
    #R = #R – 0.01
    #Z = #Z + 0.01
DONE
```

На рисунке 3.2.2 показана зависимость возвращаемого результата от аргумента «angle».

TAN angle

Единица измерения аргумента «angle»: градусы.

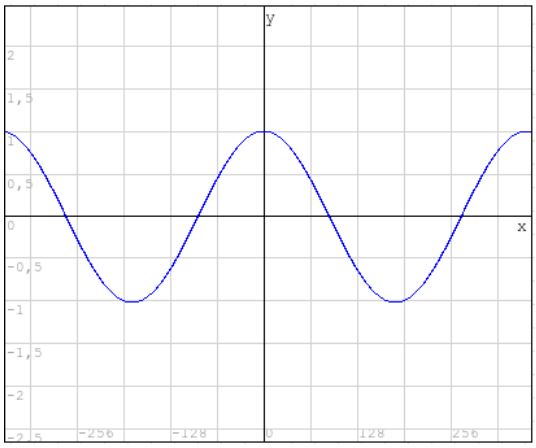


Рис. 3.2.2: График оператора cos

Возвращает тангенс угла «angle» в диапазоне [-1.0; +1.0].
Результат: значение типа FLOAT.

ASIN value

Возвращает арксинус угла аргумента «value».
Результат: значение типа FLOAT.

ACOS value

Возвращает арккосинус угла аргумента «value».
Результат: значение типа FLOAT.

ATAN value

Возвращает арктангенс угла аргумента «value».
Результат: значение типа FLOAT.

Таблица 3.2.1: Виды арифметических операций

Операция	Типы данных	Тип результата	Описание оператора
a + b	a: INT, b: INT	INT	Сумма целых чисел
a + b	a или b — не INT	FLOAT	Сумма чисел
a - b	a: INT, b: INT	INT	Разность целых чисел
a - b	a или b — не INT	FLOAT	Разность чисел
a * b	a: INT, b: INT	INT	Произведение целых чисел
a * b	a или b — не INT	FLOAT	Произведение чисел
a / b	a: INT, b: INT	INT	Целочисленное деление
a / b	a или b – не INT	FLOAT	Деление чисел
a % b	Любые	INT	Остаток от целочисленного деления
a ^ b	Любые	FLOAT	Возведение в степень a^b
a MAX b	a: INT, b: INT	INT	Максимальное из двух целых
a MAX b	a или b — не INT	FLOAT	Максимальное из двух значений
a MIN b	a: INT, b: INT	INT	Минимальное из двух целых
a MIN b	a или b — не INT	FLOAT	Минимальное из двух значений

Таблица 3.2.2: Математические функции

Пример оператора	Действие
#A = SQRT #B	$A = \sqrt{B}$
#A = CBRT #B	$A = \sqrt[3]{B}$
#A = ROUND #B	Округление B
#A = FIX #B	Округление B вниз
#A = FUP #B	Округление B вверх
#A = TRUNC #B	Отсечение дробной части у B
#A = MODF #B	Получение дробной части B
#A = ABS #B	$A = \lvert B \rvert$
#A = RND #B	Случайное число. $A \in [0; B]$
#A = LN #B	$A = \ln B$
#A = LOG10 #B	$A = \log_{10} B$
#A = EXP #B	$A = e^B$
#A = BCD #B	Перевод из BIN в BCD
#A = BIN #B	Перевод из BCD в BIN
#A = SIN #B	$A = \sin B$
#A = COS #B	$A = \cos B$
#A = TAN #B	$A = \mathrm{tg}\, B$
#A = ASIN #B	$A = \arcsin B$
#A = ACOS #B	$A = \arccos B$
#A = ATAN #B	$A = \mathrm{arctg}\, B$

3.2.2. Логические операторы

Операции сравнения

Макроязык предусматривает стандартный набор операторов сравнения. В таблице 3.2.3 приведён список операций сравнения.

NOT value

Возвращает обратное для «value» значение. Если «value» равен 0, возвращается значение 1, иначе возвращается значение 0.

Результат: значение типа INT.

ISSET value

Проверяет, было ли задано значение «value». В случае, если в качестве «value» была передана переменная, возвращает 1, если ей было задано значение (она существует и проинициализирована). В противном случае возвращает 0. Если «value» — константа (численная или строковая), возвращается значение 1.

Результат: значение типа INT.

value1 AND value2

Логическая операция «и» над значениями аргументов «value1» и «value2». Значения аргументов автоматически приводятся к значениям 0 или 1. Ниже приведена таблица истинности.

value1 OR value2

Логическая операция «или» над значениями аргументов «value1» и «value2». Значения аргументов автоматически приводятся к значениям 0 или 1. Ниже приведена таблица истинности.

Таблица 3.2.3: Операции сравнения

Синтаксис		Описание
a > b	a GT b	Возвращает 1, если $a > b$, иначе — 0
a >= b	a GE b	Возвращает 1, если $a \geqslant b$, иначе — 0
a < b	a LT b	Возвращает 1, если $a < b$, иначе — 0
a <= b	a LE b	Возвращает 1, если $a \leqslant b$, иначе — 0

Таблица 3.2.4: Таблица истинности операции AND

Значение «value1»	Значение «value2»	Результат
0	0	0
0	1	0
1	0	0
0	0	1

Таблица 3.2.5: Таблица истинности операции OR

Значение «value1»	Значение «value2»	Результат
0	0	0
0	1	1
1	0	1
0	0	1

value1 XOR value2

Логическая операция «исключающее или» над значениями аргументов «value1» и «value2». Значения аргументов автоматически приводятся к значениям 0 или 1. Ниже приведена таблица истинности.

Таблица 3.2.6: Таблица истинности операции XOR

Значение «value1»	Значение «value2»	Результат
0	0	0
0	1	1
1	0	1
0	0	0

3.2.3. Приоритет операций

Все операторы, используемые в арифметических и логических выражениях, имеют приоритет выполнения относительно других. В таблице 3.2.7 представлена группировка операций по приоритету. Чем выше группа в таблице, тем выше приоритет операции. Например, операция умножения будет выполнена раньше, чем операция сложения. А операции сложения будут выполнены раньше, чем логическая операция AND.

Операторы, у которых только один аргумент, исполняются сразу же.

Пример. Порядок вычислений

```
#X = [ 5 + 2 * 2 * 5 + 45 / 3 ] / 4 ;Операция в скобках приоритетнее
( 1. 2 * 2 = 4 )
( 2. 4 * 5 = 20 )
( 3. 45 / 3 = 15 )
( 4. 5 + 20 = 25 )
( 5. 25 + 15 = 40 )
( 6. 40 / 4 = 10 )
PRINT "#X: " #X ;Выведет на экран число 10
```

Пример. Приоритет операции при использовании оператора условия

```
#X = 5
#Y = 100
IF [ #X < 5 + 1 AND #Y > 101 - 1 ] THEN
    PRINT "Условие истинно." ; Данное сообщение будет показано
ENDIF
```

Таблица 3.2.7: Иерархия приоритетов по

Группы операций
NOT, унарный минус
MIN, MAX, ^
*, /, %, MOD
+, -
>, <, >=, <=, GT, LT, GE, LE
=, ==, !=, NE
AND, OR, XOR
INTERSECT

3.2.4. Условные оператор IF

Формат полного оператора условия IF…THEN…ELSE…ENDIF:

```
IF условие THEN
    ;операторы, выполняющиеся, если условие не равно нулю
ELSE
    ;операторы, выполняющиеся, если условие равно нулю
ENDIF
```

Полный оператор условия делится на две части: блок IF…THEN… (если условие истинно) и блок ELSE… (если условие ложно). В блоке IF…THEN… указываются операторы, выполняющиеся, если условие истинно (не равно нулю). В блоке ELSE… операторы выполняются, если условие ложно (равно нулю).

Формат неполного оператора условия IF…THEN…ENDIF:

```
IF условие THEN
    ;операторы, выполняющиеся, если условие не равно нулю
ENDIF
```

Неполный оператор условия не имеет блока ELSE…. Его можно использовать, когда нет необходимости выполнять какие-то действия, если условие не выполнилось, а также если в случае истинного условия необходимо выполнить более одного оператора.

Формат сокращённого оператора условия IF…THEN…:

```
IF условие THEN ... ;можно указать только один оператор либо кадр
```

Сокращённый оператор условия не требует указания оператора, завершающего блок IF. Однако, в случае истинного условия, можно выполнить только один произвольный оператор (либо кадр), следующий за служебным словом THEN в той же самой строке.

Условие можно представлять из себя:

- выражение в квадратных скобках;
- переменную (перед переменной можно указать знак минус);
- имя оси станка (для получения текущей координаты);
- число (может быть как INT, так и FLOAT);
- строку (в двойных кавычках).

Перед проверкой условия оно будет преобразовано в целочисленный тип INT.

3.2.5. GOTO — оператор безусловного перехода

Формат:

```
GOTO label
```

Аргументы команды:

label — метка, по которой будет происходить переход. Представляет из себя положительное число типа INT (например: 1, 2, 3, 10, 20, 100, 110, 200, 9999 и т.д.).

Безусловный переход к определённой метке предназначен для пропуска команд, организации циклов и повторного исполнения кода с новыми параметрами.

Метка должна находиться в той же самой программе/подпрограмме, в которой находится и оператор GOTO. Если в качестве метки задать отрицательное число либо несуществующую метку, то будет выдано сообщение об ошибке, а управляющая программа сбросится.

Метки должны стоять в начале каждого кадра, который необходимо пометить. Традиционный формат управляющих программ предполагает наличие в каждой строке метки с соответствующим номером. В данной системе УЧПУ рекомендуется ставить метки только в тех кадрах, где они необходимы для перехода к ним с помощью оператора GOTO либо для использования специфичными подпрограммами.

Пример. Нелинейное исполнение программы с использованием меток

```
1  G0 X0
2  #X = 0 ;Начальное значение переменной цикла
3  N100 X10 ;Помеченный кадр. Начало цикла
4  X20
5  #X += 1
6  IF [#X < 4] TH GOTO 100 ;Конец цикла. Значения #X: 0, 1, 2, 3. Далее
       переход на метку N100
7  GOTO 120 ;Переход на метку N120
8  G0 Y10
9  Y20
10 N110 Y30 ;Помеченный кадр
11 Y-10
12 EXIT ;На этой строке программа завершит работу
13 X-20
14 N120 G0 ;Помеченный кадр
15 X-10
16 GOTO 110 ;Переход на метку N110
```

Использование меток можно запретить с помощью параметра N3040.

Это может понадобиться, если необходимо, чтобы программисты системы УЧПУ не смогли писать слишком запутанные и сложные для понимания программы.

Оператор GOTO MAIN

Стандартный оператор GOTO производит переход к метке только в рамках текущей программы. При реализации подпрограмм может возникнуть необходимость перейти из подпрограммы на метку в основной программе. Для этого предназначен оператор GOTO MAIN.
Формат:

```
GOTO MAIN label
```

Аргументы команды:

label — метка, по которой будет происходить переход. Представляет из себя положительное число типа INT (например: 1, 2, 3, 10, 20, 100, 110, 200, 9999 и т.д.).

Действие оператора аналогично работе оператора GOTO за тем исключением, что подпрограмма (либо набор подпрограмм) завершается и возобновляется работа основной программы, начиная с метки «label».

3.2.6. Операторы циклов

Цикл WHILE

Формат цикла:

```
WHILE условие DO
    ;тело цикла
DONE
```

WHILE — цикл с предусловием. В качестве условия может выступать любое выражение, возвращающее значение типа INT. Цикл будет выполняться, пока условие будет возвращать значение, отличное от нуля. Если условие изначально равно нулю, цикл исполняться не будет.
Пример. Цикл, исполняющийся 5 раз

```
1 #X = 0
2 WHILE [#X < 5] DO
3     PRINT "Итерация цикла №" #X
4 DONE
5 PRINT "Цикл завершился."
```

Цикл L ... ENDL

Формат цикла:

```
L__
    ; тело цикла
ENDL
```

Цикл-счётчик. Исполняется указанное количество раз. Текущее значение счётчика цикла (самой большей вложенности) показывается на главном окне в строке состояния.

Аргументы цикла:

- **L** - количество повторов (итераций) цикла (целое положительное число типа INT).

3.2.7. Отладочные операторы

PRINT ...

Оператор предназначен для отправки сообщений (уведомлений). Удобное средство для отладки управляющих программ. Все отправленные сообщения можно просмотреть в окне «Журнал». В качестве аргументов могут выступать строки, числа, переменные и выражения, разделённые между собой пробелами.

Пример. Использование

```
1 PRINT ”Hello World!” ;Hello World!
2 PRINT ”Hello” ”World!” ;Hello World!
3 PRINT ”Hello” ”World” ”!” ;Hello World !
```

ERROR ...

Оператор предназначен для отправки сообщений об ошибках. Все отправленные сообщения можно просмотреть в окне «Журнал». В качестве аргументов могут выступать строки, числа, переменные и выражения, разделённые между собой пробелами.

Пример. Использование

```
1 ERROR ”Не указан обязательный параметр цикла X.”
```

WARN ...

Оператор предназначен для отправки сообщений с предупреждениями. Все отправленные сообщения можно просмотреть в окне «Журнал». В качестве аргументов могут выступать строки, числа, переменные и выражения, разделённые между собой пробелами.
Пример. Использование

```
1 WARN "Не указан необязательный параметр цикла A. Используется значение
    по умолчанию."
```

3.3 Подпрограммы

3.3.1. Пользовательские подпрограммы

- Каждый из циклов представляет из себя отдельный файл, являющийся подпрограммой.
- Название файла представляет из себя числовое значение в диапазоне 1000...9999 без расширения либо с расширением «.nc».
- Привязка названия файла цикла к G-коду и его атрибуты указываются через параметры программы.

Атрибуты подпрограммы

Параметры программы с N3200 по N3220 позволяют определить атрибуты цикла. Каждый из параметров программы может запрограммировать один цикл. Атрибуты перечисляются последовательно через двоеточие:

1. номер G-кода, при запуске которого будет исполняться цикл;
2. название файла цикла в виде десятичного числа;
3. тип цикла, допустимые значения:
    - 0 — простая подпрограмма,
    - 1 — контурный цикл,
    - 2 — модальный цикл, действующий только в рамках текущей программы/подпрограммы,
    - 3 — глобальный модальный цикл, распространяющийся и на подпрограммы);

4. флаг, разрешающий использование одного стека переменных на подряд идущие вызовы подпрограммы (0 — запретить, 1 — разрешить).

Примеры.

- N3200 установлен в «101:1000:1». Контурный цикл, находящийся в файле с названием «1000.nc». Вызывается через команду G101. Должен быть обязательно закончен с помощью ключевого слова CYCLEEND.
- N3201 установлен в «102:1001». Подпрограмма, находящаяся в файле с названием «1001.nc». Вызывается через команду G102.
- N3202 установлен в «103:4357:0». Подпрограмма, находящаяся в файле с названием «4357». Вызывается через команду G103.
- N3203 установлен в «105:4358:0:1». Подпрограмма, находящаяся в файле с названием «4358». Вызывается через команду G105. При повторном вызове в той же самой программе стек переменных подпрограммы сохраняется.
- N3204 установлен в «104:1233:2». Модальный цикл, находящийся в файле с названием «1233». Задаётся через команду G104.
- N3205 установлен в «105:1277:3». Модальный цикл, находящийся в файле с названием «1277». Цикл будет применяться и к подпрограммам. Задаётся через команду G104.

3.3.2. Аргументы подпрограмм

Имеются **три типа указания аргумента**.

- В **типе I** указания аргумента (таблица 3.3.1) используются основные буквы, кроме G, L, O, N и P, каждая один раз.
- В **типе II** указания аргумента (таблица 3.3.2) используются буквы A, B и C, каждая один раз, а также используются I, и K до десяти раз.
- В **типе III** указания аргумента можно использовать любые символьные названия аргументов, состоящие из букв латинского алфавита и знака подчёркивания. В подпрограмме переданные таким образом аргументы будут доступны как локальные переменные с такими же названиями.

Тип указания аргумента определяется автоматически согласно используемым буквам. Типы указания аргумента можно комбинировать, но такой

Таблица 3.3.1: Тип указания аргумента I

Адрес	Переменная	Адрес	Переменная	Адрес	Переменная
A	#1 или #A	I	#4 или #I	T	#20 или #T
B	#2 или #B	J	#5 или #J	U	#21 или #U
C	#3 или #C	K	#6 или #K	V	#22 или #V
D	#7 или #D	M	#13 или #M	W	#23 или #W
E	#8 или #E	Q	#17 или #Q	X	#24 или #X
F	#9 или #F	R	#18 или #R	Y	#25 или #Y
H	#11 или #H	S	#19 или #S	Z	#26 или #Z

Таблица 3.3.2: Тип указания аргумента II

Адрес	Переменная	Адрес	Переменная	Адрес	Переменная
A	#1 или #A	J_2	#8 или #E	K_4	#15
B	#2 или #B	K_2	#9 или #F	I_5	#16
C	#3 или #C	I_3	#10	J_5	#17 или #Q
I_1	#4 или #I	J_3	#11 или #H	K_5	#18 или #R
J_1	#5 или #J	K_3	#12	I_6	#19 или #S
K_1	#6 или #K	I_4	#13 или #M	J_6	#20 или #T
I_2	#7 или #D	J_4	#14	K_6	#21 или #U

способ указания аргументов необходимо использовать осторожно, т.к. указанный по II типу аргумент может переопределить указанный ранее по I типу (и наоборот).

Нижние индексы у букв I, J, K в программе не указываются. Они обозначают порядковый номер повторного написания буквы в качестве аргумента.
Пример. Передача аргументов в подпрограмму «2000.nc»

```
1 G65 P2000 X10 A[20 + 5] I2 J3 K5 I7 J9 K11 WORK5 GAIN[SIN 40]
```

Пример. Проверка аргументов в подпрограмме «2000.nc»

```
PRINT #X ”==” #24 ;тип I, выведет ”10 == 10”
PRINT #A ”==” #1 ;тип I, выведет ”25 == 25”
PRINT #I ”==” #4 ;тип II, выведет ”2 == 2”
PRINT #J ”==” #5 ;тип II, выведет ”3 == 3”
PRINT #K ”==” #6 ;тип II, выведет ”5 == 5”
PRINT #D ”==” #7 ;тип II, выведет ”7 == 7”
PRINT #E ”==” #8 ;тип II, выведет ”9 == 9”
PRINT #F ”==” #9 ;тип II, выведет ”11 == 11”
PRINT #WORK ;тип III, выведет ”5”
PRINT #GAIN ;тип III, выведет ”0.642788”
```

3.3.3. G65 — вызов подпрограммы из файла

При помощи функции G65 можно вызывать пользовательские подпрограммы из файлов. Данный G-код не отправляется на отработку в интерполятор и заменяется исполнением файла, на который ссылается.

Формат вызова:

```
G65 P__ L__ ...
```

Аргументы функции:

- **P** — имя подпрограммы (число в диапазоне 1000...9999)
- **L** — количество повторов подпрограммы (целое положительное число типа INT).
- Список аргументов, которые можно передавать в подпрограмму, указан в разделе 3.3.2.

Пример. Подпрограмма 1001.nc

```
O1001

G0 Z[#Z]
G91 G0 X#X
G90 G1 Z0 F100
Z[#Z/2] F200
G0 Z[#Z]
```

Пример. Вызов подпрограммы 1001.nc

```
G0 X0 Y0
G65 P1001 L10 X1 Z20 ;запуск на исполнение подпрограммы 1001.nc 10 раз с
   передачей параметров
```

3.3.4. G66 — модальный цикл из файла

При помощи функции G66 можно вызывать пользовательскую подпрограмму из файлов после каждого геометрического кадра. Данный G-код не отправляется на отработку в интерполятор и заменяется исполнением файла, на который ссылается.

Формат вызова:

```
G66 P__ L__ ...
```

Аргументы функции:

- **P** — имя подпрограммы (число в диапазоне 1000 ...9999)
- **L** — количество повторов подпрограммы (целое положительное число типа INT).
- Список аргументов, которые можно передавать в подпрограмму, указан в разделе 3.3.2.

Модальный цикл исполняется в той же строке, в которой указан, а также после каждого последующего геометрического кадра, пока не будет отменён командой G67.

Пример. Подпрограмма 1233.nc

```
O1233
G91 G1 Z#Z F200
Z-#Z F400
G90
```

Пример. Вызов подпрограммы 1233.nc

```
G90 G0 X0 Y0 Z0
G66 P1233 L10 Z10 ;модальный цикл, выполняющий сверление на 10 мм
G0 X1 ;перемещение в точку X=1, затем вызов G66
G0 X2 ;перемещение в точку X=2, затем вызов G66
G0 X3 ;перемещение в точку X=3, затем вызов G66
G67 ;отмена модального цикла G66
```

3.4 Контурные циклические подпрограммы (G...ENDG)

Контурные циклы представляют из себя подпрограммы, обрабатывающие набор кадров (контур или траекторию движения). Набор кадров указывается между началом контурного цикла и его окончанием (ключевое слово ENDG).

Все строки, указанные внутри цикла, преобразуются в набор кадров (кеш кадров), который передаётся в подпрограмму цикла. Подпрограмма может отправлять кадры в неизменном виде либо формировать свои собственные на основе расположенных в переданном наборе кадров.

Формат контурного цикла:

```
G__ ... ;начало контура цикла
    ...
ENDG ;конец контура цикла
```

Список аргументов, которые можно передавать в подпрограмму, указан в разделе 3.3.2.

Для назначения какого-либо G-кода в качестве контурного цикла необходимо в параметрах N3200...N3299 указать соответствие номера G-кода и подпрограммы, которую он будет исполнять. Все аргументы, передаваемые в подпрограмму, можно использовать согласно типу указания аргумента. Они могут выступать в роли параметров контурного цикла.

Для организации цикла обработки набора кадров в подпрограмме цикла можно использовать операторы изменения указателя на текущий кадр: FRAMEGFIRST, FRAMEGLAST, FRAMEGNEXT, FRAMEGPREV. Описание функций и переменных, которые могут быть полезны при создании контурных циклов, приведено в разделах 3.5.3 и 3.5.4 соответственно.

Пример. Прямой обход кадров

```
#B = FRAMEGFIRST ;Получаем первый кадр
WHILE [#B] DO
    ;Обработка текущего кадра
    #B = FRAMEGNEXT ;Переходим на следующий кадр
DONE
```

Пример. Обратный обход кадров

```
#B = FRAMEGLAST ; Получаем последний кадр
WHILE [#B] DO
    ;Обработка текущего кадра
    #B = FRAMEGPREV ;Переходим на предыдущий кадр
DONE
```

3.5 Служебные операторы и переменные

3.5.1. Служебные функции

UNSET #variable

Команда освобождения переменной «#variable». Команда сбрасывает флаг установленного значения у переменной. После исполнения данной команды функция ISSET, применённая к переменной «#variable» вернёт значение 0 (переменной не было установлено значение). А попытка считать значение переменной может вернуть неопределённое значение.

Пример. Освобождение переменной

```
1 PRINT [ISSET #X] ;выведет 0
2 #X = 1 ;устанавливаем значение
3 PRINT [ISSET #X] ;выведет 1
4 UNSET #X ;освобождаем переменную
5 PRINT [ISSET #X] ;выведет 0
```

WAIT

Команда синхронизации интерпретатора с интерполятором. Данная команда приостанавливает работу интерпретатора до тех пор, пока интерполятор не исполнит все кадры, указанные до данной команды.

Следует учитывать, что на момент окончания работы команды WAIT буфер кадров будет пуст, таким образом произойдёт полное торможение программы до 0-й скорости аналогично работе команды G09.

SYNC

Команда синхронизации осей между интерполятором и интерпретатором. Действие команды аналогично команде WAIT, но после исполнения всех предшествующих кадров в интерпретатор заносятся текущие координаты и изменённые смещения по осям.

Команду SYNC можно использовать в тех случаях, где в ходе исполнения программы может произойти ручное (либо от электроавтоматики) перемещение по осям, либо после выполнения модулем ПЛК автоматической привязки инструмента или детали. В таком случае интерпретатор возьмёт для своих расчётов новые координаты.

Пример. Смещение инструмента от ПЛК с синхронизацией и без

```
;Без синхронизации:
G90 G0 X0 Y0
M37 ;условная команда, ПЛК сместила инструмент в позицию X=20
G91 X1 ;перемещение произойдёт в координату X=1

;С синхронизацией:
G90 G0 X0 Y0
M37 ;условная команда, ПЛК сместила инструмент в позицию X=20
SYNC
G91 X1 ;перемещение произойдёт в координату X=21
```

PUSHSTATE

ВНИМАНИЕ! Экспериментальная функция.

Функция PUSHSTATE сохраняет состояние исполняемой программы в динамически увеличивающийся стек. Под состоянием программы понимаются все установленные на момент вызова флаги G-кодов.

POPSTATE

ВНИМАНИЕ! Экспериментальная функция.

Функция POPSTATE восстанавливает состояние программы из стека. Состояние программы предварительно должно быть сохранено с помощью команды PUSHSTATE.

SET_AXIS axis_num axis_value

Единица измерения для «axis_value»: мкм.

Устанавливает «axis_value» в качестве значения перемещения по оси «axis_num».

Пример. Создание кадра двумя способами

```
SET_AXIS 1 10000
SET_AXIS 2 20000
G1 F100
( Равносильно следующей строке при условии, что X – ось 1, Y – ось 2 )
G1 X10 Y10 F100
```

SET_FLAG flag_num

Устанавливает флаг с номером «flag_num».

Пример. Использование.

```
SET_FLAG 91
( равносильно следующей строке )
G91
```

SET_R radius_value

Единица измерения для «radius_value»: мкм.

Устанавливает в качестве радиуса для конструируемого кадра значение «radius_value».

Пример. Использование.

```
SET_R 10000
G2 X10 Y10
( равносильно следующей строке )
G2 X10 Y10 R10
```

SET_P rounds_value

Единица измерения для «rounds_value»: ед.

Устанавливает полное количество оборотов для круговой интерполяции конструируемого кадра в значение «rounds_value».

Пример. Использование.

```
SET_P 5
G2 X10 Y10 R10
( равносильно следующей строке )
G2 X10 Y10 R10 P5
```

SET_F feed_value

Единица измерения для «feed_value»: мкм/мс.

Устанавливает в качестве подачи для конструируемого кадра значение «feed_value».

Пример. Использование.

```
SET_F [ 100 * 1000 / [ 60.0 * 1000.0 ] ]
G1 Z10
( равносильно следующей строке )
G1 Z10 F100
```

SET_LINE line_no

Устанавливает номер строки конструируемого кадра в значение «line_no», чтобы на экране подсвечивалась не текущая строка, а другая (например, можно подсветить комментарий). Иначе будет подсвечена строка, в которой кадр объявлен.

При использовании функции необходимо быть осторожным, т. к. изменение индицируемой строки может привести оператора в замешательство, а также может затруднить определение текущего состояния обработки. Следует учитывать, что добавление новой строки может сместить нумерацию, в результате чего понадобится переназначать номера строк.

Пример. Подсветка строки комментария при исполнении кадров

```
;Перемещаемся к заготовке
G0 Z10
G0 X0 Y0
;Стачивание угла заготовки
G1 Z0 F50
SET_LINE 4 ;устанавливаем для следующего кадра номер индицируемой строки
X1
SET_LINE 4 ;устанавливаем и для каждого следующего кадра в рамках
   операции
Y1
SET_LINE 4
X2
SET_LINE 4
Y2
SET_LINE 4
X3
SET_LINE 4
Y3
Z10 ;в этом кадре будет индицирована уже данная строка
G0 X0 Y0
```

POSTOWORK axis_num pos

В качестве результата возвращает рабочее значение координаты «pos» оси с номером «axis_num». Результат зависит от режима G90/G91. Если задан флаг G90, то функция возвращает значение «pos». Если задан флаг G91, то в качестве результата функция возвращает значение «pos», прибавленное к текущей рабочей координате.

POSTOABS axis_num pos

В качестве результата возвращает абсолютное значение координаты «pos» оси с номером «axis_num». Интерпретация координаты pos зависит от режима G90/G91.

ISDIAMETRIC axis_num

Возвращает 1, если оси с номером «axis_num» диаметральная, и 0 — если ось радиусная.

EXIT

Завершение работы управляющей программы. Все последующие кадры обрабатываться не будут.

HALT

Аварийное завершение работы управляющей программы. Все последующие кадры обрабатываться не будут. Буфер кадров очищается, текущий кадр аварийно завершает свою работу.

Рекомендуется использовать внутри подпрограмм в случаях, если в подпрограмму были переданы некорректные аргументы, либо не были переданы необходимые аргументы.

3.5.2. Служебные переменные

#N1..#N9999

Только для чтения.

Параметры УЧПУ. Для получения значения параметра после буквы N указывается номер параметра.

Пример. Получение количества осей

```
#COUNT = #N1000 ;Количество осей
PRINT #COUNT ;Выведет на экран количество осей в системе
```

#N[index]

Только для чтения.

Параметры УЧПУ. Вместо index должен указываться номер параметра.

Пример. Получение названий всех осей по их номерам

```
#COUNT = #N[1000] ;Количество осей
#I = 1;
WHILE [#I <= #COUNT] DO
    ( Выведет на экран номера и названия всех осей )
    PRINT "Ось: " #I "; название:" #N[5000 + #I * 100]
DONE
```

#R1..#R255

Тип данных: FLOAT.

Ячейки памяти УЧПУ, предназначенные для постоянного хранения информации. Ячейки сохраняются в файл на носителе информации в формате JSON. Сохранение происходит в момент старта программы и в момент окончания программы. В случае непредвиденного завершения работы системы УЧПУ (отключение питания), последние изменения переменных могут быть утеряны.

#I1..#I255

Тип данных: BYTE.

Ячейки входов УЧПУ.

#U1..#U255

Тип данных: BYTE.

Ячейки выходов УЧПУ.

#_AXIS_COUNT

Только для чтения.

Общее количество осей станка.

Например, для простого фрезерного станка со шпинделем, у которого есть оси X, Y и Z, значение данной переменной будет 4 (3 оси + 1 шпиндель).

#G_FLAG[group_id]

Только для чтения.

Массив групп G-флагов. Возвращает текущий G-код для группы «group_id», где вместо «group_id» необходимо записать номер группы G-кодов (0…17).

Пример. Сохранение и восстановление состояние флага G90/G91

```
#G9X = #G_FLAG[3] ;сохраняем в переменную #G9X значение флага группы 3 - систему координат (90 или 91)
G91 G0 X100 Y100
G1 X2 Y0
G1 X0 Y2
G#G9X ;восстанавливаем систему координат согласно значению в переменной #G9X
```

#G_AXIS[axis_num]

Только для чтения.

Единица измерения: мкм.

Массив значений по осям отправляемой команды. Указывает на значения перемещений по каждой из осей, где «axis_num» — порядковый номер оси (нумерация с 1-цы). По умолчанию пуст, если туда не было занесено значение.

Пример. Конструирование кадра

```
SET_AXIS 1 1000
SET_AXIS 1 [#G_AXIS[1] + 1000] ;Сдвигаем кадр по оси номер 1 на 1 мм
G1 ;Отправляем кадр
```

#G_XI, #G_YI и #G_ZI

Только для чтения.

Переменные #G_XI и #G_YI содержат номера осей, образующих выбранную плоскость (по G17, G18, G19, G220). Переменная #G_ZI содержит номер оси, перпендикулярной к выбранной плоскости.

#G_CI

Только для чтения.

Переменная #G_CI содержит номер оси, являющейся текущим шпинделем.

#G_F

Только для чтения.

Единица измерения: мкм/мс или об/мс.

Подача обрабатываемого кадра. Следует учитывать, что данное значение нельзя напрямую указывать в качестве аргумента F, т.к. его единицы измерения отличаются от задаваемых. Данное значение можно использовать для задания скорости через оператор SET_F.

#_AXIS_POS[axis_num]

Только для чтения.

Единица измерения: мкм.

Массив значений по осям текущих расчётных рабочих координат.

«axis_num» — порядковый номер оси (нумерация с 1-цы). При старте программы значения массива соответствуют значениям текущих рабочих координат.

Пример. Получение текущей координаты по оси X

```
1 G54 G90 G0 X10
2 PRINT #_AXIS_POS[1] ;Выведет на экран 10000.0
```

#_AXIS_ABS_POS[axis_num]

Только для чтения.

Единица измерения: мкм.

Массив значений по осям текущих расчётных абсолютных координат. «axis_num» — порядковый номер оси (нумерация с 1-цы). При старте программы значения массива соответствуют значениям текущих абсолютных координат. В абсолютных координатах диаметральные оси вычисляются как радиусные (т. е. после исполнения кадра «G90 G53 G1 X10» значение #_AXIS_ABS_POS[1] будет равно 5).

Пример. Получение абсолютного значения текущей координаты по оси X

```
1 G54 G91 G0 X10
2 PRINT #_AXIS_ABS_POS[1] ;Выведет на экран абсолютную координату оси
```

#_SUP

Только для чтения.

Номер текущего суппорта. Нумерация с 1.

Пример. Перемещение по разным осям, в зависимости от суппорта

```
IF [#_SUP = 1] THEN
    T11.1 M6 ;выбираем инструмент 1-го суппорта
ENDIF
IF [#_SUP = 2] THEN
    T21.1 M6 ;Выбираем инструмент 2-го суппорта
ENDIF
```

#_R_CORR

Только для чтения.
Величина текущей коррекции на радиус.

#_CHL

Только для чтения.
Номер текущего суппорта. Нумерация с 1.

Пример. Перемещение по разным осям, в зависимости от суппорта

```
IF [#_CHL = 1] THEN
    G0 X10 ;Перемещаемся по оси X первого канала
ENDIF
IF [#_CHL = 2] THEN
    G0 U10 ;Перемещаемся по оси U второго канала
ENDIF
```

#_T

Только для чтения.
Номер текущего инструмента.

Пример. Смена инструмента в зависимости от текущего

```
G28 ;перемещаемся в точку смены инструмента
IF [#_T = 1] THEN
    T2
ELSE
    IF [#_T = 2] THEN
        T3
    ENDIF
ENDIF
```

#_TB

Только для чтения.
Номер текущей привязки инструмента.

#_TR

Только для чтения.
Единица измерения: мм.
Радиус текущего инструмента.

#_T_ANGLE

Только для чтения.
Единица измерения: градусы.
Угол текущего инструмента.

#_T_TYPE

Только для чтения.
Единица измерения: нет.
Тип текущего инструмента.

#_TH

Только для чтения.
Единица измерения: мм.
Длина текущего инструмента.

#_TKD

Только для чтения.
Единица измерения: мм.
Значение текущего корректора на радиус.

#_TKH

Только для чтения.
Единица измерения: мм.
Значение текущего корректора на длину.

#_CL

Только для чтения.

Текущее значение счётчика повторов текущей подпрограммы. Нумерация с 0. При использовании в основной программе всегда возвращает число 0.

#_CL_COUNT

Только для чтения.

Количество повторов текущей подпрограммы. При использовании в основной программе всегда возвращает число 1.

#PLANE_I[daxis_num]

Только для чтения.

Массив, содержащий номера осей, совпадающих по направлению с декартовыми осями. daxis_num — номер декартовой оси:

1 — декартова ось X,

2 — декартова ось Y,

3 — декартова ось Z.

3.5.3. Служебные функции подпрограмм

SET_G_DIRECT value

Если в качестве значения «value» было передано значение «1», то все кадры, формирующиеся после данной команды, будут отправлены напрямую в буфер кадров.

Если в качестве значения «value» было передано значение «0», то все кадры, формирующиеся после данной команды, не будут отправлены напрямую в буфер кадров. Кадр можно добавить в текущий набор (кеш) кадров с помощью оператора FRAMEGPUSH для дальнейшей обработки.

FRAMEGPUSH LAST

Сохраняет последний отправленный на исполнение кадр в текущий набор (кеш) кадров.

FRAMEGFIRST

Устанавливает указатель на начало набора обрабатываемых кадров. Возвращает 1, если кадр существует, 0 — если набор кадров пуст.

FRAMEGLAST

Устанавливает указатель на конец набора обрабатываемых кадров. Возвращает 1, если кадр существует, 0 — если набор кадров пуст.

FRAMEGNEXT

Устанавливает указатель на следующий кадр из набора обрабатываемых кадров. Возвращает 1, если кадр существует, 0 — если в наборе больше не осталось кадров.

FRAMEGPREV

Устанавливает указатель на предыдущий кадр из набора обрабатываемых кадров. Возвращает 1, если кадр существует, 0 — если в наборе перед текущим больше нет кадров.

FRAMEGSEND

Отправляет на исполнение текущий кадр из набора обрабатываемых кадров. Возвращает 1, если кадр отправлен, 0 — если произошла ошибка.

FRAMEGMEM

Возвращает указатель на текущий кадр из набора обрабатываемых кадров. Полученное значение можно далее использовать в функции FRAMEGJUMP.

FRAMEGJUMP

Осуществляет переход к запомненному ранее (с помощью функции FRAMEGME кадру из набора обрабатываемых кадров.

FRAMEGCLEAR

Очищает кеш обрабатываемых кадров, удаляя из него все ранее сохранённые (через FRAMEGPUSH либо автоматически) кадры.

FRAMEGMIN axis_num

Возвращает минимальное значение координаты по оси с номером «axis_num» в текущем кадре из набора обрабатываемых кадров.

FRAMEGMAX axis_num

Возвращает максимальное значение координаты по оси с номером «axis_num» в текущем кадре из набора обрабатываемых кадров.

FRAMEGLINE

Возвращает номер строки текущего кадра из набора обрабатываемых кадров.

SET_LINE

Устанавливает номер строки конструируемого кадра в значение номера строки обрабатываемого кадра, чтобы на экране правильно отображать, какой именно кадр мы обрабатываем. Иначе будет подсвечена строка самого цикла.

ISREPEATED

Возвращает 1, если модальная подпрограмма была вызвана после исполнения кадра, 0 — если подпрограмма запущена в рамках своего собственного вызова.

3.5.4. Служебные переменные контурных циклов

#_AXIS[axis_num]

Только для чтения.

Единица измерения: мкм.

Массив значений по осям обрабатываемого кадра. Указывает на значения перемещений по каждой из осей, где «axis_num» — порядковый номер оси (нумерация с 1-цы). Чтобы проверить, было ли выставлено перемещение по определённой оси, необходимо использовать функцию ISSET.

Пример. Создание нового кадра на основе первого из набора

```
( Получаем первый кадр )
#B = FRAMEGFIRST
IF [#B] THEN ;если кадр существует
    ( Если была указана координата X, получаем её значение )
    IF [ISSET #_AXIS[1]] THEN
        #X = #_AXIS[1]
    ELSE
        ( Если координата X не была передана, сделаем перемещение в 0 )
        #X = 0
    ENDIF
    ( выполняем перемещение )
    G1 X#X
ENDIF
```

#_AXIS_A[axis_num]

Только для чтения.

Единица измерения: мкм.

Массив координат начала обрабатываемого кадра. Указывает на значения координат начала кадра по каждой из осей, где «axis_num» — порядковый номер оси (нумерация с 1-цы).

#_AXIS_B[axis_num]

Только для чтения.

Единица измерения: мкм.

Массив конечных координат обрабатываемого кадра. Указывает на значения координат конца кадра по каждой из осей, где «axis_num» — порядковый номер оси (нумерация с 1-цы).

#_FLAG[group_id]

Только для чтения.

Массив групп G-флагов обрабатываемого кадра. Возвращает G-код для группы «group_id», где вместо «group_id» необходимо записать номер группы G-кодов (0..17).

Пример. Отправка кадров круговой интерполяции согласно текущему из набора

```
( Получаем первый кадр )
#B = FRAMEGFIRST
WHILE [#B] DO ;пока кадр существует
    #GMT = #_FLAG[2] ;получаем флаг для 2й группы, значение: 17, 18 или
   19
    #X = #_AXIS[1]
    #Y = #_AXIS[2]
    G2 G#GMT X#X Y#Y ;выставляем флаг текущей плоскости и формируем кадр
    #B = FRAMEGNEXT
DONE
```

#_XI и #_YI

Только для чтения.

Переменные #_XI и #_YI содержат номера осей, образующих выбранную плоскость (по G17, G18, G19, G220) для обрабатываемого кадра.

#_R

Только для чтения.

Единица измерения: мкм.

Радиус текущего обрабатываемого кадра.

#_F

Только для чтения.

Единица измерения: мкм/мс.

Подача обрабатываемого кадра. Следует учитывать, что данное значение нельзя напрямую указывать в качестве аргумента F, т. к. его единицы измерения отличаются.

Пример. Увеличение подачи в 2 раза

```
SET_F [#_F * 2]
```

#_FRAME_NUM

Только для чтения.

Номер кадра, заданный в управляющей программе с помощью обозначения N___

Предметный указатель