

ОГЛАВЛЕНИЕ

Глава 1. Основные положения.....	5
1.1. Этапы работы системы.....	5
1.2. Общие термины и понятия	6
1.3. Задачи, решаемые адаптером	6
1.3.1. Определение имени станка	7
1.3.2. Определение номера постпроцессора	7
1.3.3. Преобразование команд CLDATA в слова и кадры УП	7
1.3.4. Компоновка кадров управляющей программы	10
Глава 2. Этапы создания постпроцессора.....	12
2.2. Действия пользователя при написании постпроцессора	13
2.3. Распечатка файлов постпроцессора	17
2.4. Трансляция файла алгоритмов.....	18
2.5. Просмотр результатов работы постпроцессора	18
2.2. Отладка постпроцессора	19
Глава 3. Формирование паспорта станка.....	21
3.1. Действия пользователя	21
3.2. Содержимое паспорта станка	22
3.2.1. Оборудование.....	22
3.2.2. Шпиндель, подача, охлаждение.....	22
3.2.3. Инструмент.	23
3.2.4. Корректоры.	24
3.2.5. Перемещения.	25
3.2.6. Интерполяторы.....	26
3.2.7. Стандартная величина аппроксимации.....	26
3.2.8. Циклы.	26
3.2.9. Таблица перекодировки символов.....	27
3.2.10. Параметры управляющей программы.....	27
Глава 4. Формирование файла макрокоманд	28
4.1. Действия пользователя	28
4.2. Пример формирования файлов макрокоманд.....	29
Глава 5. Формирование макета кадра	31
5.1. Формат вывода.....	31
5.2. Формирование окон различных типов	34
5.3. Действия пользователя	35
Глава 6. Формирование файла алгоритмов	37
6.1. Арифметические действия и функции в алгоритмах.....	37

6.2. Команды алгоритмов.....	40
6.3. Пример работы с трансформами.....	50
6.4. Примеры работы с пользовательскими командами и циклами	52
6.4.1.Пример работы с пользовательскими командами	53
6.4.2.Пример работы с пользовательскими циклами	60
6.5. Действия пользователя	67
6.6. Пример формирования файла алгоритмов.....	68

Глава 7. Системные переменные..... 73

7.1. Координаты инструмента	73
7.2. Круговая интерполяция	75
7.3. Последующие перемещения инструмента.....	78
7.4. Совмещенные перемещения.	81
7.5. Геометрия и номер позиции инструментов	81
7.6. Включение/выключение корректоров	83
7.7. Выстой.....	83
7.8. Положение металла.....	84
7.9. Управление шпинделем	84
7.10. Управление подачей	85
7.11. Резьба	86
7.12. Учетные параметры программы, детали и станка	87
7.13. Постоянные циклы	87
7.14. Координаты безопасной позиции.....	88
7.15. Координаты точки прижима	88
7.16. Номер стола	89
7.17. Номер трубопровода СОЖ.....	89
7.18. Начало цикла.....	89
7.19. Работа с подпрограммами.....	90
7.20. Системные переменные для работы с контурами и CLData.....	91
7.21. Системные переменные для работы с пользовательскими функциями....	92
7.22. 5-ти координатные перемещения и работа с трансформами.....	92
7.23. Системные переменные для работы с элементами контуров.	93
7.24. Вспомогательные переменные	93
7.25. Пользовательские переменные	94

Глава 8. Отладчик алгоритмов 95

Приложение А. Команды CLDATA..... 97

Приложение Б. Пример паспорта станка 99

Приложение В. Пример файла макрокоманд	101
Приложение Г. Пример макета кадра	102
Приложение Д. Пример файла алгоритмов	104

Глава 1. Основные положения

1.1. Этапы работы системы

Проектирующая часть модуля ADEM CAM (процессор) готовит последовательность команд обработки в универсальном виде (CLDATA). Программа, переводящая эту последовательность команд из формата CLDATA в формат конкретной стойки ЧПУ, называется *процессором адаптации* или *адаптером*. Смотрите рис. 1.

В своей работе адаптер использует *постпроцессор* на станок и формирует на рабочем диске файл с именем PLENT.TAP, содержащий текст управляющей программы в формате ASCII.

Стойки ЧПУ работают с различными системами кодирования символов, отличающимися от формата ASCII, например: ISO, БЦК и др. Чтобы подготовить управляющую программу для загрузки в стойку ЧПУ, необходимо перекодировать каждый символ файла PLENT.TAP из формата ASCII в формат этой стойки. Это делает *перекодировщик*, который запускается автоматически после отработки адаптера. Сформированный перекодировщиком файл PROG.TAP содержит управляющую программу в формате стойки ЧПУ.

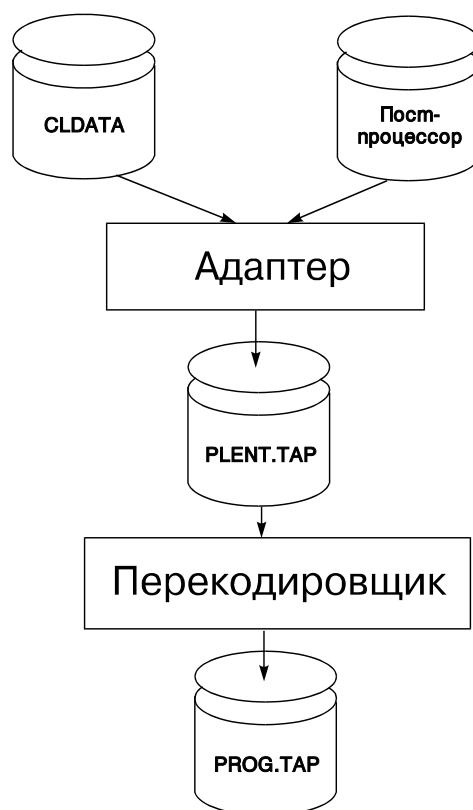


Рис. 1. Схема работы адаптера

Файлы CLDATA, PLENT.TAP и PROG.TAP являются временными файлами, при выходе из системы они уничтожаются. УП должна быть записана на диск командой **Сохранить УП как**. Создаются два файла (основное имя задается пользователем):

<имя>.TAP – УП в текстовом виде (ASCII-формат);

<имя>.TNC – УП для передачи на станок в формате, определенном в постпроцессоре.

1.2. Общие термины и понятия

CLDATA в системе ADEM называется промежуточная (процессор – адаптер) информация о траектории инструмента и технологических параметрах обработки. CLDATA состоит из последовательности команд.

Каждая команда CLDATA обозначает определенное действие, имеет свой код и может иметь параметры.

Пример:

Команда	Код	Параметры	Действие
Включить ускоренное перемещение	25	Без параметров	Включение ускоренного перемещения
Линейное перемещение	181	Координаты X, Y, Z	Перемещение инструмента в точку с координатами X, Y, Z
Включить рабочую подачу	23	Величина подачи	Включение заданной подачи

Чтобы получить управляющую программу, необходимо представить последовательность действий, содержащихся в файле CLDATA, в виде кадров управляющей программы на конкретный станок.

Два термина, известные всем программистам для станков с ЧПУ.

Слово УП (слово) – составная часть кадра УП, содержащая данные о параметре процесса обработки или другие управляющие данные.

Адрес ЧПУ (адрес) – часть слова УП, определяющая назначение следующих за ним данных этого слова.

1.3. Задачи, решаемые адаптером

Адаптер решает следующие задачи:

- Определяет имя станка, на который нужно получить управляющую программу.
- Определяет по имени станка номер постпроцессора, который будет использоваться при формировании управляющей программы.

- Преобразует команды CLDATA в слова и кадры управляющей программы.
- Компонует информацию в виде единой управляющей программы.

1.3.1. Определение имени станка

Имя станка задается командой **СТАНОК** в модуле ADEM CAM (смотрите руководство по ADEM CAM). Это имя является параметром команды CLDATA СТАНОК (код 3).

Внимание! • После изменения имени станка необходимо повторно сгенерировать CLDATA, т. е. выполнить команду **Процессор**.

1.3.2. Определение номера постпроцессора

Определение номера постпроцессора производится по имени станка, который ищется в каталоге станков. Каталог станков – текстовый файл с именем STANKI.SKR, находящийся в директории \ADEM\NCM\POSTPR вместе с постпроцессорами для станков.

Каталог содержит записи следующего формата:

| <имя станка> | <комментарий> | <номер постпроцессора> |

Имя станка, заданное командой **СТАНОК**, должно точно соответствовать имени, записанному в каталоге станков в графе <имя станка>. Если адаптер не обнаружит в каталоге заданного имени, он выдаст сообщение **“Станок не включен в каталог”**, адаптер завершит работу, управляющая программа сформирована не будет. При добавлении постпроцессора на станок, который уже есть в каталоге, нужно изменить его название (например, необходимо добавить каталог постпроцессор на станок АГПН-630, а постпроцессор на такой станок уже существует – в этом случае заносим в каталог название станка АГПН-630 v.1).

1.3.3. Преобразование команд CLDATA в слова и кадры УП

Каждой команде CLDATA поставлен в соответствие алгоритм представления ее действия в формате управляющей программы. Преобразование команды CLDATA в часть управляющей программы осуществляется в два этапа:

- Поиск по коду команды CLDATA ее алгоритма.
- Реализация найденного алгоритма.

Поиск алгоритма по коду команды CLDATA

Алгоритмы отображения действия команд CLDATA содержатся в файле алгоритмов. Этот файл является частью постпроцессора на станок и имеет имя, например, для постпроцессора с номером 222, FTTP0222.ANK. Без этого файла управляющая программа формироваться не будет, адаптер выдаст сообщение **“Нет файла алгоритмического заполнителя”**.

Если на какую-либо команду CLDATA алгоритм не будет найден, она никак не отразится в управляющей программе.

Реализация алгоритма

Алгоритм представляет собой последовательность строк следующего формата:

[IF] <условие выполнения> [ELSE] <команда алгоритма>;

IF указывает, что команда должна быть выполнена только при соблюдении условия, следующего за **IF**. Только строки с **IF** могут иметь альтернативные строки.

ELSE указывает, что данная строка является альтернативной для строки с **IF**, расположенной выше.

Пример: Алгоритм может выглядеть следующим образом:

```
.  
3->_E;  
_X->ХТ;  
_Y->УТ;  
IF E!=1 ПНКАДР;  
КАДР;  
.
```

Введем некоторые новые понятия:

Системные и пользовательские переменные

Как было отмечено выше, команда CLDATA может иметь параметры. Значения параметров присваиваются соответствующим *системным переменным* при реализации ее алгоритма. Например, при обработке алгоритма команды *Линейное перемещение* (код 181) значения координат точки текущего положения инструмента X, Y и Z присваиваются системным переменным с именами ХТ, УТ и ЗТ (Описание системных переменных смотрите в [главе 7](#)).

Кроме системных переменных существуют *пользовательские переменные*. Их имена должны начинаться со знака подчеркивания “_”, но значения пользовательских переменных определяет только разработчик анкеты.

Внимание! • При составлении алгоритма можно пользоваться только системными и пользовательскими переменными.

В приведенном выше примере ХТ и УТ – системные переменные, а _E, _X и _Y – пользовательские.

Окно кадра

Окно кадра описывает слово кадра управляющей программы и состоит из двух частей:

Символьная часть соответствует адресу (может содержать последовательность символов).

Формат вывода определяет вид выводимой числовой информации (например, максимальное количество выводимых символов, количество позиций после десятичной точки и т. д.).

Пример: G[]

G - символьная часть окна,

[] - условное обозначение формата вывода.

Макет кадра

Окна кадра в той последовательности, в которой они должны располагаться в кадре, содержатся в *файле макета кадра*. Таким образом, *макет кадра* – это структура кадра УП: взаимное расположение всех возможных слов кадра и описание каждого из них.

Этот файл является частью анкеты на станок и имеет имя, например для анкеты с номером 222, KADR0222.ANK. Без этого файла управляющая программа формироваться не будет, адаптер выдаст сообщение **“Нет макета кадра”**.

Вывод информации в окно макета кадра

Алгоритм формирует УП, выводя информацию в окна макета кадра. Эту информацию он получает из CLDATA посредством системных переменных.

Команда алгоритма, выводящая информацию в окно макета кадра, имеет следующий формат:

<номер окна в макете кадра> → <выводимая информация>;

Как видно из контекста команды, для вывода информации в окно необходимо указать его порядковый номер в макете кадра.

Пример: Инструмент находится в точке с координатами X=10, Y=20, Z=30.

Макет кадра:

N[]G[]G[]X[]Y[]Z[]M[]....

Отработанные команды алгоритма:

4→ХТ ;

5→УТ ;

6→ЗТ ;

2→1 ;

Сформированная часть кадра:

G1X10.Y20.Z30.

1.3.4. Компоновка кадров управляющей программы

Как было сказано выше, управляющая программа формируется выводом информации в окна макета кадра. При этом выводимая информация поступает сначала в буфер формируемого кадра. Затем содержимое этого буфера передается в УП и образует там отдельный кадр. Передача эта происходит, например, при отработке команды алгоритма КАДР. Подробно механизм формирования кадров УП описан в [Главе 8](#).

При формировании управляющей программы выполняются следующие правила:

Автоматическая нумерация кадров

Автоматическая нумерация кадров осуществляется через окно типа НОМЕР КАДРА (смотрите [раздел 6.2](#)). Формат вывода этого окна содержит интервал нумерации.

Автоматическое формирование конца кадра

Автоматически формируется окно типа КОНЕЦ КАДРА.

Сохранение информации

Кадр управляющей программы условно делится на части, каждая из которых определяет включение/выключение какой-либо функции или ее параметры. Эти функции могут быть *альтернативными*, когда функция отменяет действие предыдущей (например, функции G0, G1, G2, G3 в стойке FANUC), а могут быть совместно работающими, т. е. могут размещаться в одном кадре

(например, G1, F, M в стойке FANUC). Для контроля размещения этих функций в кадрах введены два понятия:

Тип окна

Тип окна – это тип функции станка, реализуемый через данное окно (вспомогательная функция, подготовительная функция и т. д.).

Номер группы окна

Номер группы окна – это номер группы *альтернативных* функций соответствующего типа.

Рассмотрим ситуацию. В какое-либо окно заносится информация (например, в окно G[] заносится 1). При этом возможны два случая:

- а) Формируемый кадр не содержит альтернативной функции, тогда он дополняется заносимой в окно информацией (G1).
- б) Формируемый кадр уже содержит альтернативную функцию (например, G2). В этом случае информация, содержащаяся в формируемом кадре, выведется в УП отдельным кадром, начнется формирование нового кадра.

Поддержка модальности

Функции в кадрах управляющей программы бывают двух типов: *модальные* и *локальные*. Модальные действуют до их отмены альтернативной функцией, локальные действуют в пределах одного кадра. Для учета этой особенности составления управляющей введено понятие *модальность окна*.

Модальность окна

Если указано, что окно действует модально, адаптер запоминает последнее выведенное в это окно значение. Когда по алгоритму вновь приходит команда вывести информацию в это окно, адаптер сравнивает последнее выведенное и выводимое значения. Если они одинаковы, информация не выводится.

Гашение пустых кадров

Адаптер осуществляет контроль на наличие информации при выводе кадра. Если по алгоритму отрабатывается команда КАДР (Конец кадра), а в формируемом кадре нет информации, команда КАДР игнорируется.

Глава 2. Этапы создания постпроцессора

2.1. Состав постпроцессора

Постпроцессор состоит из четырех частей:

Паспорт станка	Общие данные по станку и правилам программирования.
Макрокоманды	Информация об обработке адаптером таких команд CLDATA, для реализации которых необходимо выполнить несколько команд CLDATA.
Макет кадра	Структура кадра управляющей программы: взаимное расположение всех возможных окон кадра и описание каждого из них.
Алгоритмы	Алгоритмы представления команд CLDATA в виде кадров и слов управляющей программы.

Что такое *макет кадра* и *файл алгоритмов*, мы достаточно подробно рассмотрели в [разделе 1.3](#). Дополнительную информацию о них Вы сможете найти в [главе 6](#) и [главе 7](#).

Паспорт станка

Паспорт станка – это набор вопросов и возможные варианты ответов о станке и правилах программирования для него.

Примеры вопросов:

- Тип оборудования;
- Возможность программного управления охлаждением
- Наличие кругового интерполятора
- Точность аппроксимации и др.

Макрокоманды

Иногда возникает необходимость детализировать какую-либо из команд CLDATA. Наиболее часто эта ситуация возникает с командами CLDATA *Загрузить инструмент* и *Конец управляющей программы*.

Например, система закончила обработку текущего объекта в точке с координатами X=35.5, Y=70, Z=-30 (последней была команда *Линейная интерполяция*) и сгенерировала команду на загрузку нового инструмента для обработки следующего технологического объекта, то есть файл CLDATA содержит команды:

```
.  
ИДИ/ТОЧ 35.5 70 -30  
ИНСТР/2  
.
```

Возникла ситуация, когда отработавший инструмент еще находится в зоне обработки, а команда CLDATA требует его заменить. В этом случае, чтобы сменить инструмент, нужно отвести его в позицию смены инструмента на ускоренной подаче, выключить при отводе корректор на длину, выключить

шпиндель, выключить охлаждение и только тогда провести смену инструмента. Поэтому команде *Загрузить инструмент* необходимо поставить в соответствие следующую последовательность команд:

1. Включить ускоренное перемещение (код 25)
2. Выключить корректор по оси Z (код 709)
3. Отвести инструмент (код 28)
4. Выключить охлаждение (код 700)
5. Выключить шпиндель (код 701)
6. Загрузить инструмент (код 35)

Таким образом, команда CLDATA *Загрузить инструмент* будет заменена макрокомандой, а шесть перечисленных команд будут подкомандами этой макрокоманды.

Макрокоманды находятся в файле макрокоманд, который является необязательной частью анкеты и имеет имя, например, для анкеты с номером 222, MCOM0222.ANK.

2.2. Действия пользователя при написании постпроцессора

Вход в модуль ADEM GPP

Запустите программу **ADEM GPP – генератор постпроцессоров**. Нажмите кнопку **ПУСК** на «Панели задач» операционной системы Windows. В дополнительном меню выберите **Программы\ADEM\ADEM GPP** и нажмите левую кнопку мыши.

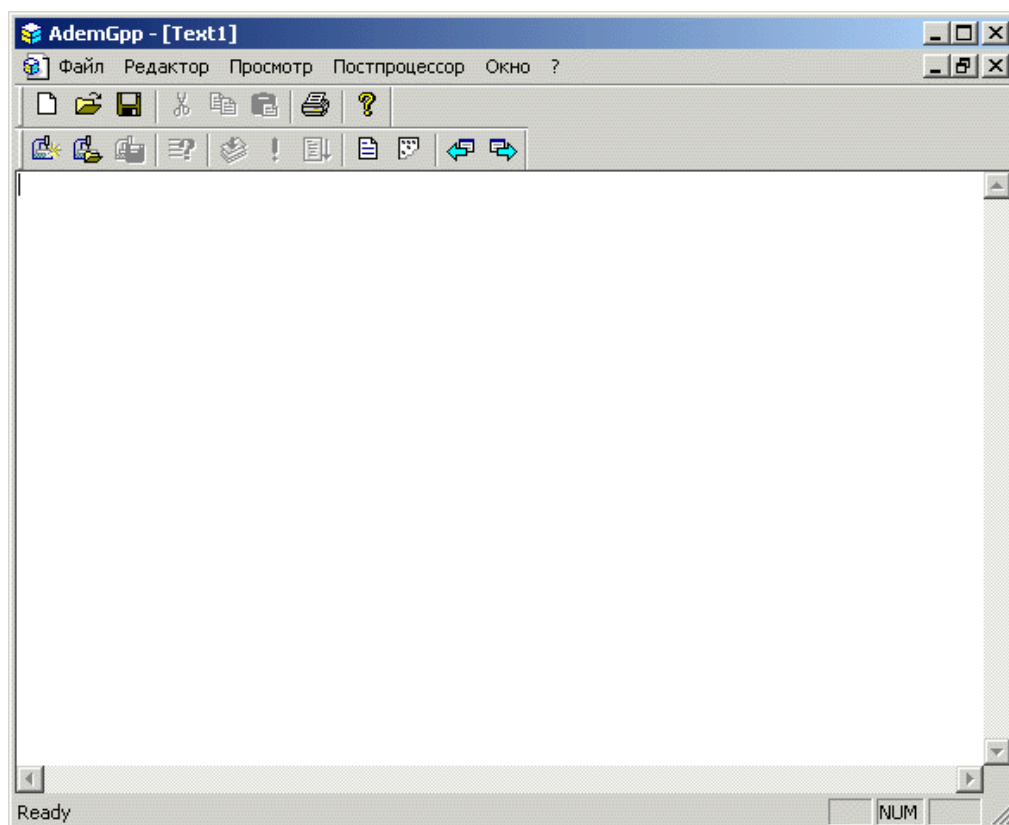


Рис. 2. Главное окно программы.

Генератор постпроцессоров позволяет выполнить следующие действия:

Создать новый постпроцессор

Нажмите кнопку **Создать постпроцессор**  Откроется диалог «Новый постпроцессор».

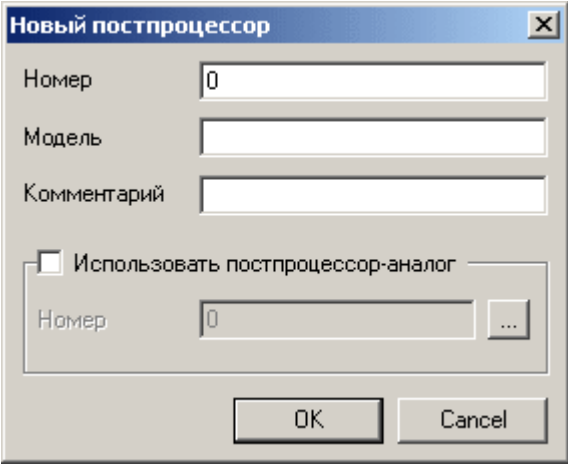


Рис. 3. Новый постпроцессор.

В поле «Номер» введите номер постпроцессора

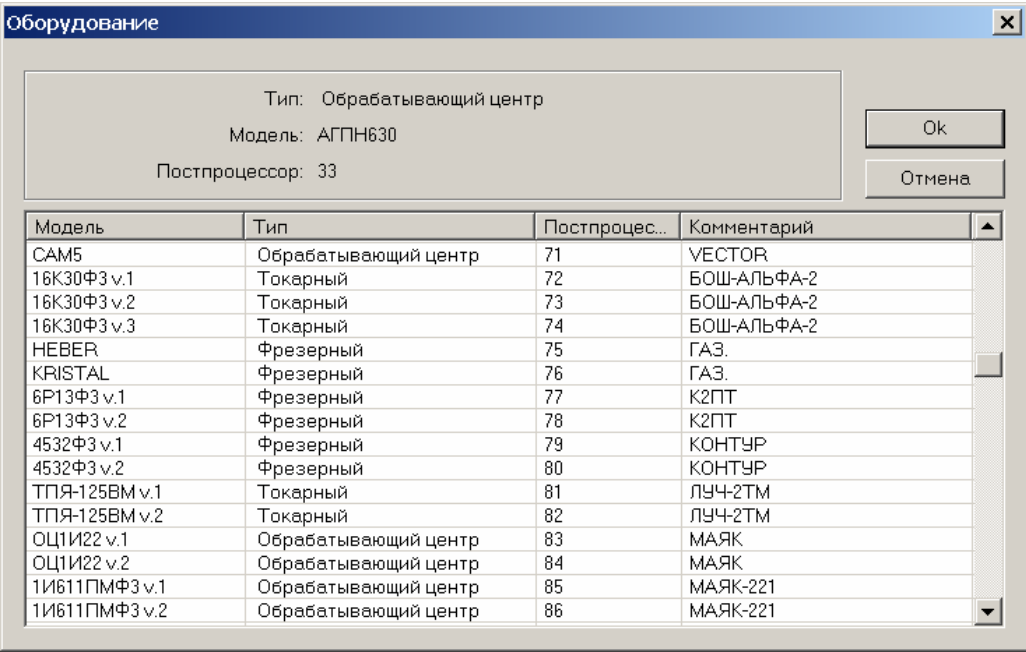
В поле «Модель» введите модель станка

В поле «Комментарий» введите комментарий

Для создания постпроцессора можно использовать аналогичный постпроцессор, разработанный до этого. Для этого установите флажок «Использовать постпроцессор-аналог» и в поле «Номер» введите номер аналогичного постпроцессора. Можно выбрать постпроцессор из списка, если нажать кнопку возле поля «Номер». После заполнения диалога нажмите клавишу **ОК**.

Открыть постпроцессор

Нажмите кнопку **Открыть постпроцессор**  . Появится диалог «Открыть постпроцессор».



Модель	Тип	Постпроцес...	Комментарий
CAM5	Обрабатывающий центр	71	VECTOR
16K30Ф3 v.1	Токарный	72	БОШ-АЛЬФА-2
16K30Ф3 v.2	Токарный	73	БОШ-АЛЬФА-2
16K30Ф3 v.3	Токарный	74	БОШ-АЛЬФА-2
HEBER	Фрезерный	75	ГАЗ.
KRISTAL	Фрезерный	76	ГАЗ.
6P13Ф3 v.1	Фрезерный	77	K2ПТ
6P13Ф3 v.2	Фрезерный	78	K2ПТ
4532Ф3 v.1	Фрезерный	79	КОНТУР
4532Ф3 v.2	Фрезерный	80	КОНТУР
ТПЯ-125BM v.1	Токарный	81	ЛУЧ-2ТМ
ТПЯ-125BM v.2	Токарный	82	ЛУЧ-2ТМ
ОЦ1И22 v.1	Обрабатывающий центр	83	МАЯК
ОЦ1И22 v.2	Обрабатывающий центр	84	МАЯК
1И611ПМФ3 v.1	Обрабатывающий центр	85	МАЯК-221
1И611ПМФ3 v.2	Обрабатывающий центр	86	МАЯК-221

Рис. 4. Открыть постпроцессор.

Откроется диалог выбора постпроцессоров. Из списка выберите нужный постпроцессор и нажмите **ОК**. Если известен номер вашего постпроцессора, то введите номер в поле «Номер постпроцессора» и нажмите **ОК**.

Сохранить постпроцессор

Нажмите кнопку **Сохранить постпроцессор**  Система сохранит активный постпроцессор.

Заполнить паспорт станка

Нажмите кнопку **Параметры** . Появится диалог «Постпроцессор».

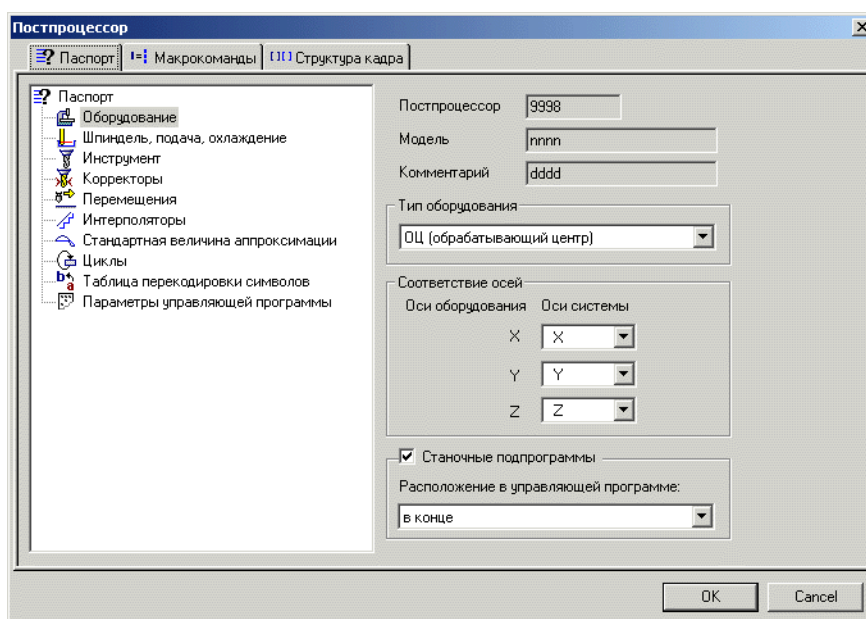



Рис. 5. Заполнение паспорта.

Процесс заполнения паспорта подробно описан в Главе 3.

Сформировать макрокоманду

Нажмите кнопку **Параметры**  Появится диалог «Постпроцессор». Выберите закладку **Макрокоманды**.

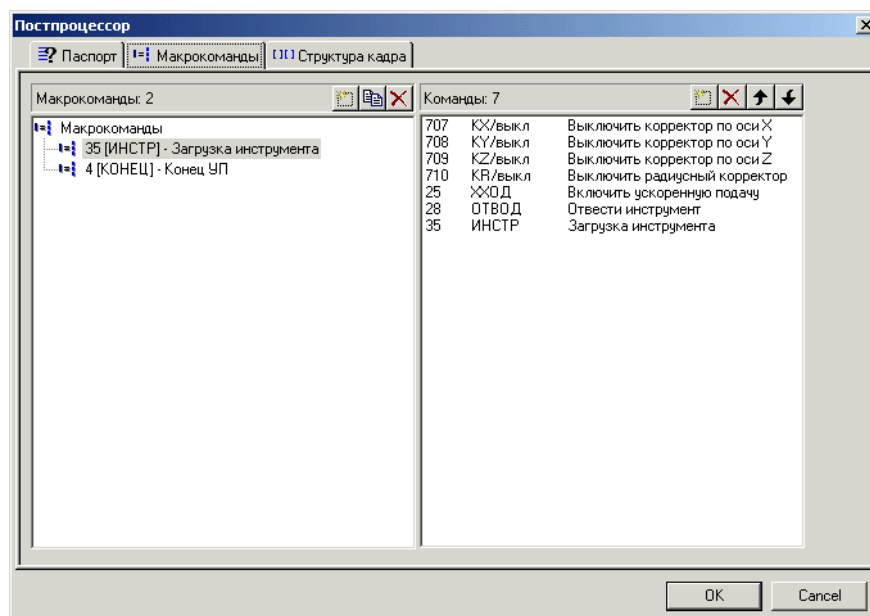


Рис. 6. Формирование макрокоманды.

Процесс формирования макрокоманды подробно описан в [Главе 4](#).

Сформировать макет кадра

Нажмите кнопку **Параметры** . Появится диалог «Постпроцессор». Выберите закладку **Структура кадра**.

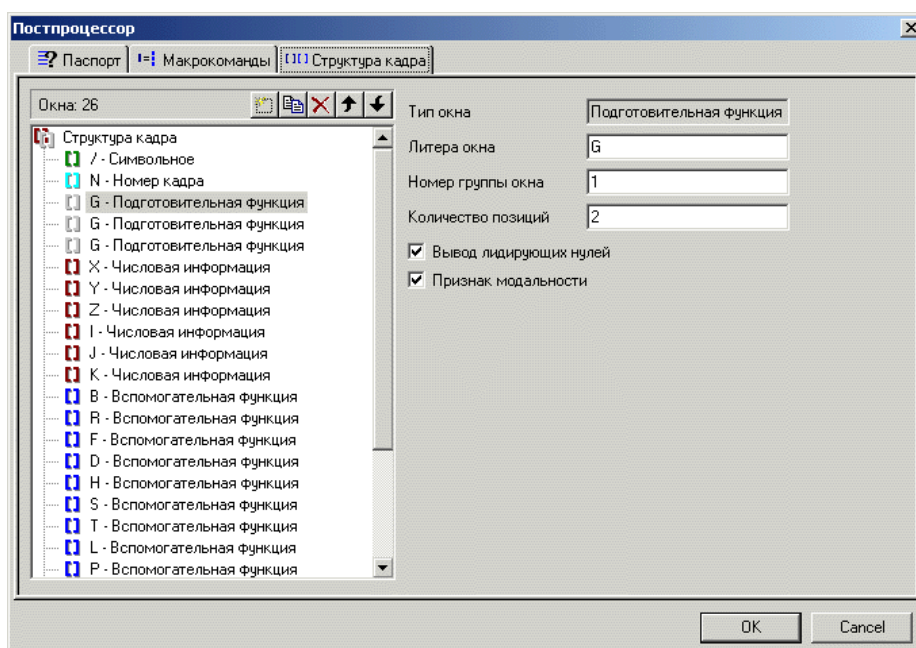



Рис. 7. Формирование макета кадра.

Процесс формирования макета кадра подробно описан в [Главе 5](#).


Сформировать новый файл алгоритма

Нажмите кнопку **Создать** . Откроется окно с пустым текстовым файлом.


Открыть алгоритм

Нажмите кнопку **Открыть**  и выберите файл алгоритма. Система откроет выбранный файл.

Сохранить изменения в алгоритме

Нажмите кнопку **Сохранить**  Система сохранит изменения в текущем алгоритме.

Работа с буфером обмена

Нажмите кнопку **Вырезать**  чтобы вырезать текст и поместить в буфер.

Нажмите кнопку **Копировать**  чтобы поместить текст в буфер.

Нажмите кнопку **Вставить**  чтобы поместить текст из буфера в окно алгоритма.

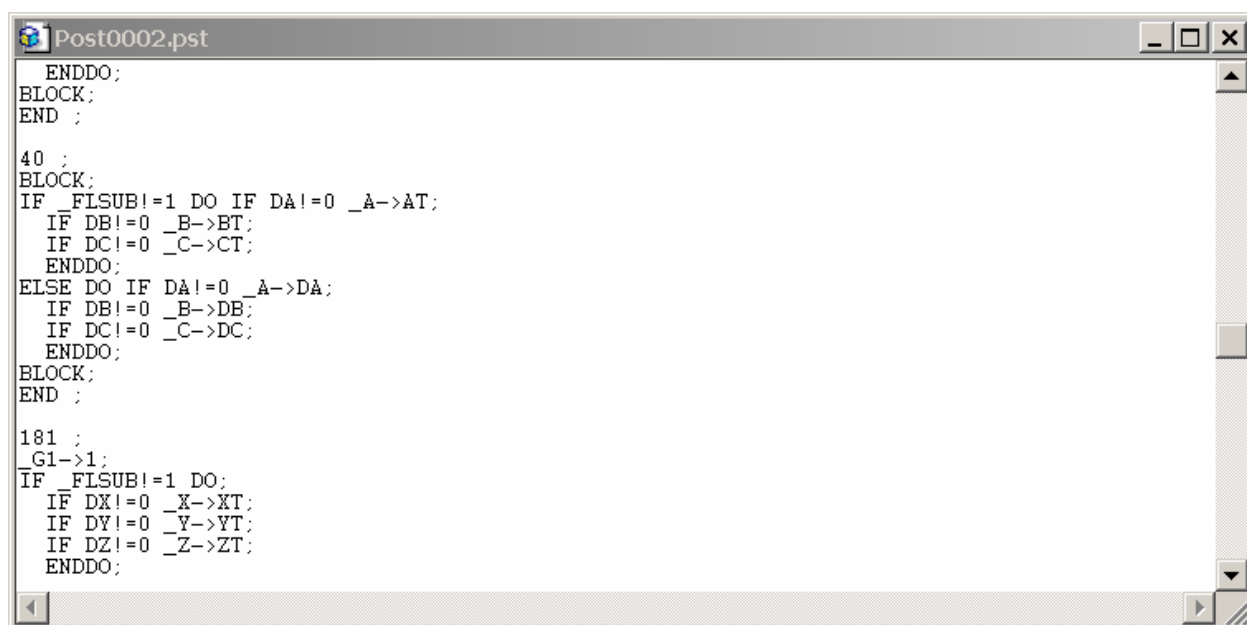



Рис. 8. Текст алгоритма.


Процесс формирования алгоритмов подробно описан в [Главе 6](#).

2.3. Распечатка файлов постпроцессора

- Для настройки вашего принтера выберите из меню **Файл** ⇒ **Настройка печати**.
- Для просмотра печатаемого текста выполните из меню **Файл** ⇒ **Просмотр печати**.
- Распечатайте полученный текст, выбрав из меню **Файл** ⇒ **Печать** либо нажмите кнопку **Печать** .

2.4. Трансляция файла алгоритмов

При помощи трансляции текстовый файл алгоритма переводится в коды, понятные машине. Для выполнения трансляции:

- Создайте новый или откройте старый постпроцессор. В текстовом редакторе появится текст алгоритма, в который можно вносить изменения.
- Нажмите кнопку **Трансляция** .

Если в тексте не обнаружено ошибок, появится сообщение **Трансляция выполнена успешно**.

Если в тексте есть ошибки, на экране появится сообщение.

Сообщение имеет следующий формат:

<номер строки> : Команда <код команды> Ошибка в строке <текст строки>

<номер строки>	Номер строки алгоритма, содержащей ошибку.
<код команды>	Код команды CLDATA, в алгоритме которой обнаружена ошибка.
<текст строки>	Текст строки, содержащей ошибку.

В окне текстового редактора система подсветит строку с ошибкой. Исправьте ее и повторите трансляцию.


2.5. Просмотр результатов работы постпроцессора

Для того, чтобы проверить этот постпроцессор, Вы должны войти в модуль ADEM CAM, командой **СТАНОК** установить станок, для которого создали постпроцессор, запустить (возможно, повторно) **Процессор**, затем **Адаптер** и просмотреть получившуюся УП.

Если в постпроцессоре оказались ошибки, вновь вернитесь в ADEM GPP и измените его.

ADEM GPP позволяет создавать УП, просмотреть текст управляющей программы и текст CLDATA при помощи следующих команд:

Выполнить Адаптер

Нажмите кнопку **Адаптер** . Система переведет CLDTA в УП при помощи текущего постпроцессора (в ADEM CAM должен быть установлен тот же постпроцессор).

Просмотреть CLDATA

Нажмите кнопку **CLDATA** . Система откроет окно с текстом CLDATA.

Просмотреть управляющую программу.

Нажмите кнопку **Управляющая программа** . Система откроет окно с текстом УП.

Просмотреть предыдущее окно

Нажмите кнопку **Предыдущее окно** . Система покажет предыдущее окно.


Просмотреть следующее окно

Нажмите кнопку **Следующее окно** . Система покажет следующее окно.

2.2. Отладка постпроцессора

Существует возможность строчной отладки алгоритмов команд CLDATA (это также способ проверки постпроцессора).

Отрабатываются команда за командой, при этом Вы имеете возможность следить за значениями всех переменных, формируемым кадром и сформированной управляющей программой, оперативно изменить алгоритм. Для отладки необходимо:

- В модуле ADEM CAM установите имя станка, соответствующее отлаживаемому постпроцессору и подготовьте CLDATA.
- Войдите в ADEM GPP и откройте отлаживаемый постпроцессор.
- Нажмите кнопку **Отладка**  ..
- Запустится отладчик алгоритма, описание работы с ним смотрите в [Главе 8](#).

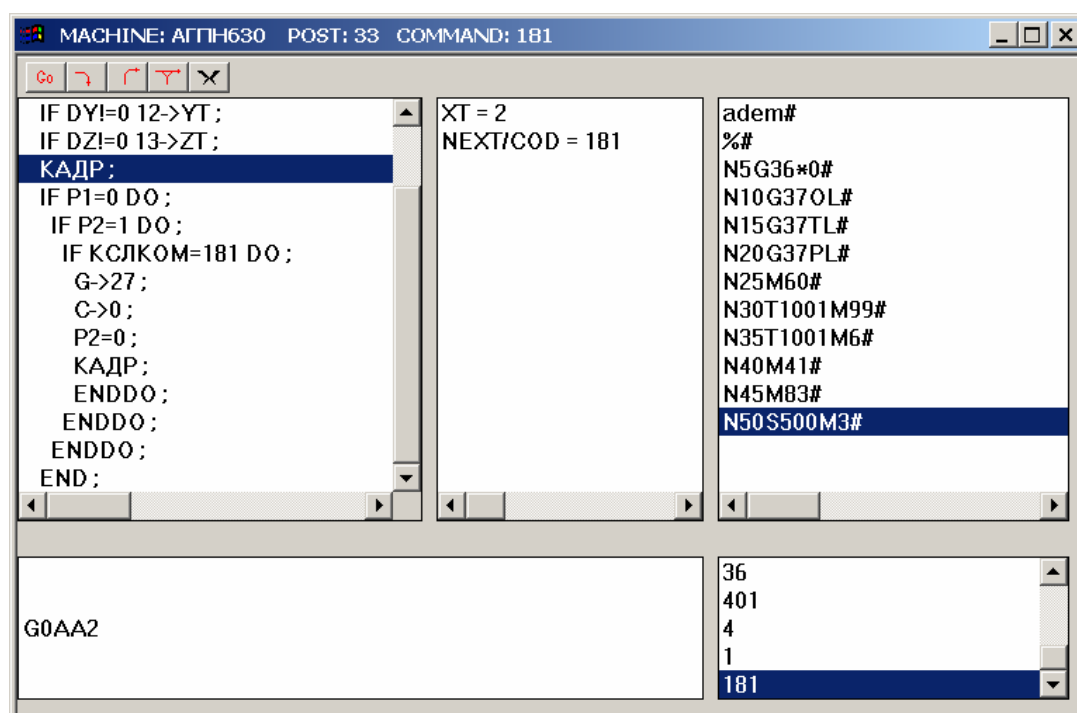


Рис. 9. Отладчик.


Примечание:

- После изменения имени станка необходимо пересчитать CLDATA, т. е. выполнить команду **Процессор** в ADEM CAM.

Глава 3. Формирование паспорта станка

Паспорт станка является частью постпроцессора и имеет имя, например, для постпроцессора с номером 222, FANK0222.ANK. Без этого файла управляющая программа формироваться не будет, адаптер выдаст сообщение **“Нет файла паспорта станка”**.

3.1. Действия пользователя

Запустите модуль подготовки и отладки постпроцессоров. Для этого нажмите кнопку **Параметры** . Откроется окно с активизированной закладкой **Паспорт**.

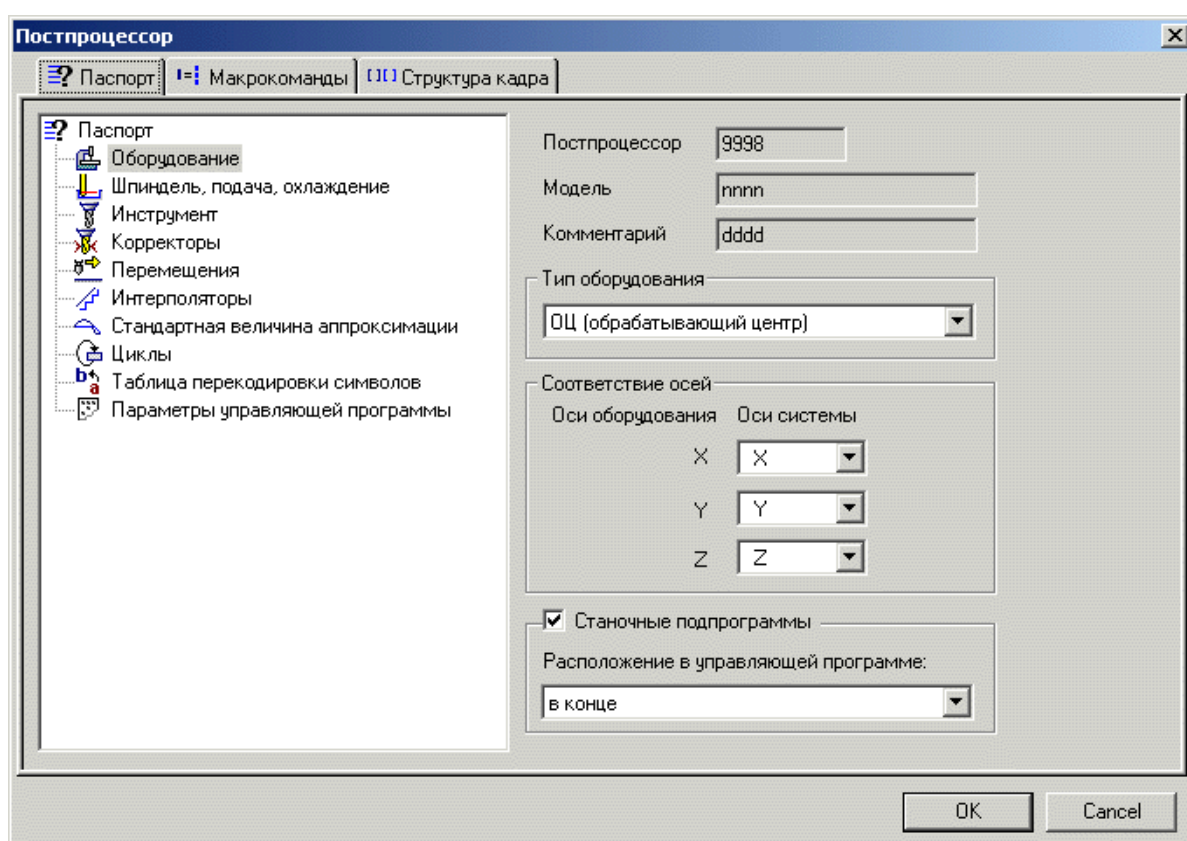


Рис. 10. Окно паспорта станка.

Всю необходимую информацию для заполнения паспорта Вы можете найти в документации к Вашему станку: в паспорте станка или инструкции по программированию.

При корректировке паспорта система выдает на экран дерево разделов паспорта. Пользователь выбирает какой-либо раздел и вводит информацию по нему.

3.2. Содержимое паспорта станка

3.2.1. Оборудование

1. Тип оборудования

- Токарное
- Обработывающие центры
- Фрезерное
- Сверлильное
- Прессы

2. Соответствие осей систем координат детали и станка

Определение соответствия между координатными осями оборудования и системы.

Пример: Задано следующее соответствие осей:

Оси оборудования	Оси системы
X	Y
Y	-Z
Z	X

В CLDATA существует перемещение в точку с координатами
X=100, Y=200, Z=300.

Адаптер формирует перемещение в точку с координатами
X=200, Y=-300, Z=100.

3 Станочные подпрограммы

Определение положения текста подпрограммы в управляющей программе. Для включения этой функции необходимо установить соответствующий флажок и выбрать из списка расположение подпрограммы.

3.2.2. Шпиндель, подача, охлаждение.

1. Управление шпинделем

Из соответствующего списка можно выбрать два варианта:

- Вместе с перемещением
- Немедленно после получения команды

Если выбран вариант **В одном кадре с перемещением**, адаптер после получения команды CLDATA на включение шпинделя “придерживает” ее до появления команды линейной или круговой интерполяции.

Если выбрано **Немедленно после получения команды**, адаптер формирует команду на включение шпинделя сразу после получения команды из CLDATA.

2. Управление подачей

Существует два варианта:

- Вместе с перемещением
- Немедленно после получения команды

Если выбран первый вариант, адаптер после получения команды на включение подачи “придерживает” ее до появления из CLDATA команды линейной или круговой интерполяции и помещает ее в одном кадре с перемещением.

Если выбрано **Немедленно после получения команды**, адаптер формирует команду на включение подачи сразу.

3. Управление охлаждением

3.1. Возможность программного управления охлаждением.

Реализовано ли управление охлаждением на станке? Если нет, все команды, связанные с включением/выключением охлаждения, игнорируются. Для включения программного управления охлаждением установите соответствующий флажок

3.2. Место выдачи в УП команды включения/выключения СОЖ.

Существует три варианта:

- Немедленно после получения команды,
- Вместе с перемещением,
- Вместе с перемещением на рабочей подаче.

Если задан вариант **Немедленно после получения команды**, адаптер формирует команду на включение охлаждения сразу после получения команды из CLDATA.

Если задан вариант **Вместе с любым перемещением**, адаптер после получения команды на включение охлаждения “придерживает” ее до появления из CLDATA команды линейной или круговой интерполяции и выдает ее непосредственно перед командой на линейное или круговое перемещение.

Аналогично работает адаптер при выборе **Вместе с перемещением на рабочей подаче**, только в этом случае адаптер “придерживает” команду до линейной или круговой интерполяции на рабочей подаче.

3.2.3. Инструмент.

1. Загрузка первого инструмента в конце управляющей программы.

Если установлен флажок, то при загрузке последнего инструмента следующим инструментом будет установлен первый инструмент данной УП.

Если флажок не установлен, то при загрузке последнего инструмента номером следующего инструмента будет ноль.

2. Позиция смены инструмента

Существует восемь вариантов позиций смены инструмента:

- **В любой точке** При смене инструмента адаптер не генерирует дополнительные перемещения.
- **В плоскости НЦ по** При смене инструмента адаптер генерирует

- | | |
|-------------------------------|---|
| X | отвод в координату X начала цикла. |
| • В плоскости НЦ по Y | При смене инструмента адаптер генерирует отвод в координату Y начала цикла. |
| • В плоскости НЦ по XY | При смене инструмента адаптер генерирует отвод в координаты X и Y начала цикла. |
| • В плоскости НЦ по Z | При смене инструмента адаптер генерирует отвод в координату Z начала цикла. |
| • В плоскости НЦ по ZX | При смене инструмента адаптер генерирует отвод в координаты Z и X начала цикла. |
| • В плоскости НЦ по YZ | При смене инструмента адаптер генерирует отвод в координаты Z и Y начала цикла. |
| • В точке НЦ | При смене инструмента адаптер генерирует отвод в точку начала цикла. |

3. Фиксированные координаты отвода

Позволяет задать фиксированные координаты отвода по осям X, Y, Z. Для ввода значения необходимо установить флажок возле соответствующей координаты.

3.2.4. Корректоры.

1. Включение линейных и радиусного корректоров

1.1. Место выдачи в УП команд включения линейных корректоров.

Определяется для каждой из осей X, Y и Z.

Существует пять вариантов:

- Немедленно после получение команды
- С положительным перемещением по оси
- С отрицательным перемещением по оси
- С любым перемещением по оси
- Коррекция не предусмотрена

Если установлен флажок и выбрано **Немедленно после получения команды**, в УП команда на включение корректора выдается сразу после получения из CLDATA команды на включение корректора или команды загрузки инструмента, если в ней был указан корректор по соответствующей оси.

Если установлен флажок и задана выдача команды с каким-либо перемещением (второй, третий и четвертый варианты), команда “придерживается” адаптером до появления из CLDATA линейной или круговой интерполяции с перемещением по соответствующей оси и выдается непосредственно перед этой командой.

Если флажок не установлен, то команды CLDATA на включение корректора игнорируются.

1.2. Место выдачи в УП команды включения радиусного корректора.

Существует три варианта:

- Немедленно после получения команды
- При любом перемещении по оси

- Не предусмотрено

При установке флажка и выборе варианта **Немедленно после получения команды** команда на включение корректора выдается в УП сразу после получения этой команды из CLDATA.

Если установлен флажок и выбрано **При любом перемещении по оси**, команда “придерживается” адаптером до появления из CLDATA команды линейной или круговой интерполяции и выдается непосредственно перед этой командой.

Если флажок не установлен, то команды на включение радиусного корректора игнорируются.

3.2.5. Перемещения.

1. Плоскость холостых ходов

Предусмотрено четыре варианта:

- **Флажок не установлен** В этом случае адаптер строит отвод через координату Z безопасной позиции.
- **Плоскость XY** Устанавливается величина плоскости отвода, параллельной плоскости XY.
- **Плоскость XZ** Устанавливается величина плоскости отвода, параллельной плоскости XZ.
- **Плоскость YZ** Устанавливается величина плоскости отвода, параллельной плоскости YZ.

Примечание: • Как правило, выбирается **Флажок не установлен**. В этом случае адаптер формирует отвод, исходя из текущей ситуации.

В поле **Z координата** вводится значение высоты плоскости холостого хода.

2. Разрешенные холостые ходы

Существует четыре варианта:

- **Разрешены холостые ходы по трем координатам одновременно** Холостые ходы адаптер не контролирует.
- **Разрешены холостые ходы совместно по осям X и Y** Адаптер генерирует холостые перемещения одновременно по осям X и Y.
- **Разрешены холостые ходы совместно по осям X и Z** Адаптер генерирует холостые перемещения совместно по осям X и Z.
- **Разрешены холостые ходы совместно по осям Z и Y** Адаптер генерирует холостые перемещения одновременно по осям Z и Y.

3.2.6. Интерполяторы.

1. Линейный интерполятор

1.1. Наличие линейного интерполятора.

Если устройство ЧПУ не имеет линейного интерполятора, все линейные и круговые перемещения интерполируются отрезками, параллельными осям X, Y или Z. Для включения линейного интерполятора установите соответствующий флажок.

1.2. Максимальные перемещения по осям X, Y, Z.

Если при перемещении из точки в точку разница координат будет превышать указанную максимальную величину, адаптер разобьет это перемещение на несколько равных по длине перемещений, приращения координат в которых не будут превышать указанное здесь значение.

Если максимальное перемещение по оси равно нулю, оно считается не заданным и контролироваться системой не будет.

2. Круговой интерполятор

2.1. Наличие кругового интерполятора.

Если ЧПУ не имеет кругового интерполятора, все круговые перемещения аппроксимируются отрезками. Для включения кругового интерполятора необходимо установить соответствующий флажок.

2.2. Необходимость разбивки дуги на квадранты.

Если оборудование позволяет выполнить круговую интерполяцию только в пределах одного квадранта, а круговое перемещение в CLDATA проходит через несколько квадрантов, адаптер разобьет его на несколько перемещений, каждое из которых будет лежать в пределах одного квадранта. Для включения разбивки на квадранты необходимо установить соответствующий флажок.

2.3. Максимальный радиус интерполяции.

Если радиус кругового перемещения в CLDATA будет превышать это значение, адаптер аппроксимирует эту дугу отрезками со стандартной точностью, заданной в разделе **Стандартная величина аппроксимации**.

3.2.7. Стандартная величина аппроксимации.

С такой точностью будут при необходимости аппроксимированы отрезки и дуги.

3.2.8. Циклы.

Если существуют технологические объекты с переходами СВЕРЛИТЬ, ЦЕНТРОВАТЬ, РАЗВЕРНУТЬ, ЗЕНКЕРОВАТЬ или РАСТОЧИТЬ ОТВЕРСТИЕ, формирование CLDATA будет зависеть от информации, помещенной в этом разделе. Каждый из вышеперечисленных переходов можно представить в виде одного из постоянных циклов.

Система просит указать, какие из сверлильно-расточных циклов (по FANUC с 81 по 89) реализованы в стойке станка.

Если цикл реализован в описываемом станке, в CLDATA формируется команда цикл (код 36), параметрами которой будут являться соответствующий переходу номер цикла (NЦИКЛ) и его параметры (подача, глубина обработки и т. д.).

Если в постпроцессоре адаптер не найдет номер соответствующего цикла, процессор развернет обработку на уровне перемещений.

В ADEM существует возможность создавать пользовательские циклы. Описывать их в паспорте не нужно. Обработка пользовательских циклов происходит в соответствующем алгоритме (36 алгоритм) по их номеру.

3.2.9. Таблица перекодировки символов.

Содержит символы ASCII и байты, которые нужно вывести вместо них при перекодировке.

3.2.10. Параметры управляющей программы.

1. Код формирования ленты

Управляющая программа сначала формируется в ASCII-кодах, т. е. создается обычный текстовый файл (имя). Затем автоматически перекодировщиком этот файл перекодировка в форму, воспринимаемую станком.

Существует три варианта кодировки УП:

- **ISO четный** Перекодировка производится из ASCII в коды ISO.
- **ISO нечетный** Используется редко.
- **Произвольный** Перекодировка определяется только по постпроцессору. (Смотрите раздел паспорта **Таблица перекодировки символов.**)

2. Количество пропусков между кадрами

Устанавливается количество двоичных нулей, которые выведутся в файл при перекодировке, после каждого кадра.

3. Длина заправочной части ленты

При перекодировке в начало файла запишется такое количество двоичных нулей, при котором длина пустой ленты была бы равна заданной. (Длина задается в метрах.)

4. Начальный блок УП

Последовательность символов, которую необходимо вывести в начале управляющей программы.

5. Конечный блок УП

Последовательность символов, которую необходимо вывести после завершения формирования управляющей программы.

Глава 4. Формирование файла макрокоманд

Напомним, что файл макрокоманд содержит информацию об обработке адаптером тех команд CLDATA, для реализации которых необходимо выполнить несколько команд CLDATA.


Например, команде *Загрузить инструмент* бывает нужно поставить в соответствие команды

1. Включить ускоренное перемещение (код 25)
2. Выключить корректор по оси Z (код 709)
3. Отвести инструмент (код 28)
4. Выключить охлаждение (код 700)
5. Выключить шпиндель (код 701)
6. Загрузить инструмент (код 35)

Файл макрокоманд является необязательной частью постпроцессора и имеет имя, например, для постпроцессора с номером 222 - MCOM0222.ANK.

4.1. Действия пользователя

Запустите модуль подготовки и отладки постпроцессоров.

Нажмите кнопку **Параметры**  - появится окно **Постпроцессор**. Активизируйте закладку **Макрокоманды**.

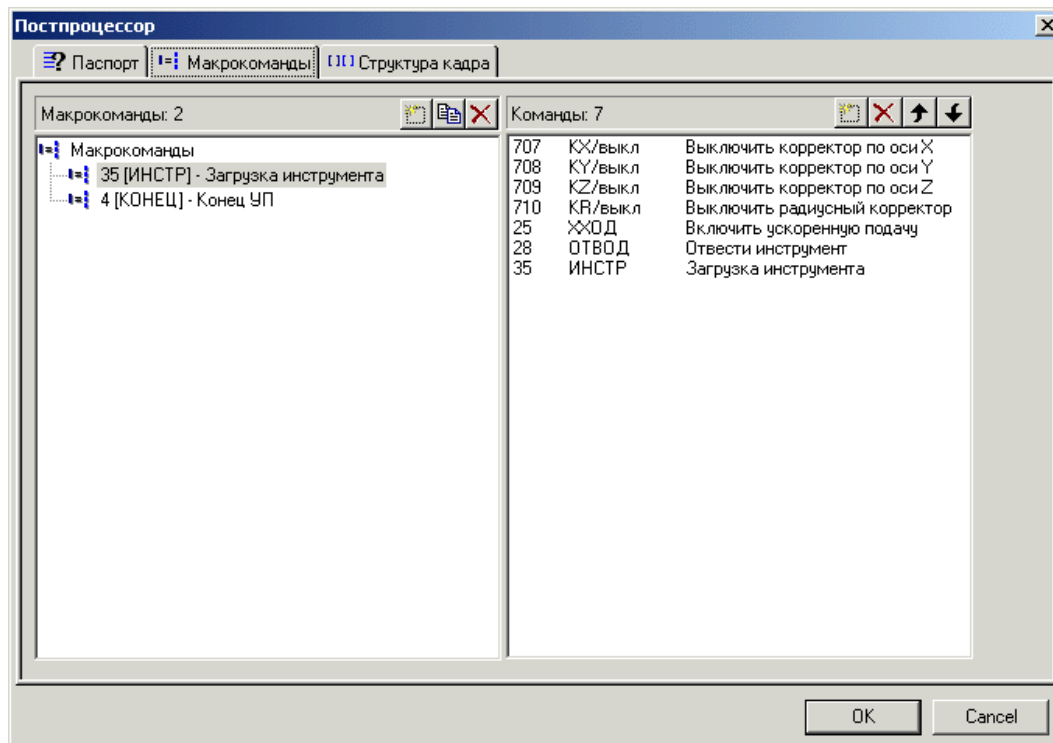


Рис. 11. Окно макрокоманды.

Закладка имеет два окна. В левом окне находятся сформированные макрокоманды, а в правом создаются подчиненные элементы. Для формирования


макрокоманды необходимо выбрать из списка в левом окне макрокоманду, которая будет включать в себя вспомогательные команды. Затем в правом окне создайте вспомогательные команды. Для работы используйте кнопки правого и левого окна. Если кнопка принадлежит левому окну, то она работает с макрокомандами. Если кнопка принадлежит правому окну, то она работает с подчиненными элементами.

Кнопка **Создать**  добавляет новый элемент из списка команд CLDATA.

Кнопка **Копировать**  копирует выделенный элемент.

Кнопка **Удалить**  удаляет выделенный элемент.

Кнопка **Переместить вверх**  перемещает вверх выделенный элемент вверх.

Кнопка **Переместить вниз**  перемещает вниз выделенный элемент вниз.

Полный список команд CLDATA с кодами смотрите в [Приложении А](#).

4.2. Пример формирования файлов макрокоманд

Пример 1.

Необходимо сформировать файл макрокоманд постпроцессора с номером 1 и занести в него планы отработки двух команд **Загрузка инструмента** и **Конец управляющей программы**.

Для левого окна при помощи кнопки **Создать** добавьте:

- **Загрузка инструмента**

Для правого окна при помощи кнопки **Создать** добавьте:

- выключить корректор X;
- выключить корректор Y;
- выключить корректор Z;
- включить ускоренную подачу;
- отвести инструмент;
- выключить шпиндель;
- загрузка инструмента.

Для левого окна при помощи кнопки **Создать** добавьте:

- **Конец управляющей программы**

Для правого окна при помощи кнопки **Создать** добавьте:

- выключить корректор X;
- выключить корректор Y;
- выключить корректор Z;
- включить ускоренную подачу;
- отвести инструмент;
- выключить СОЖ;
- выключить шпиндель;
- конец программы.

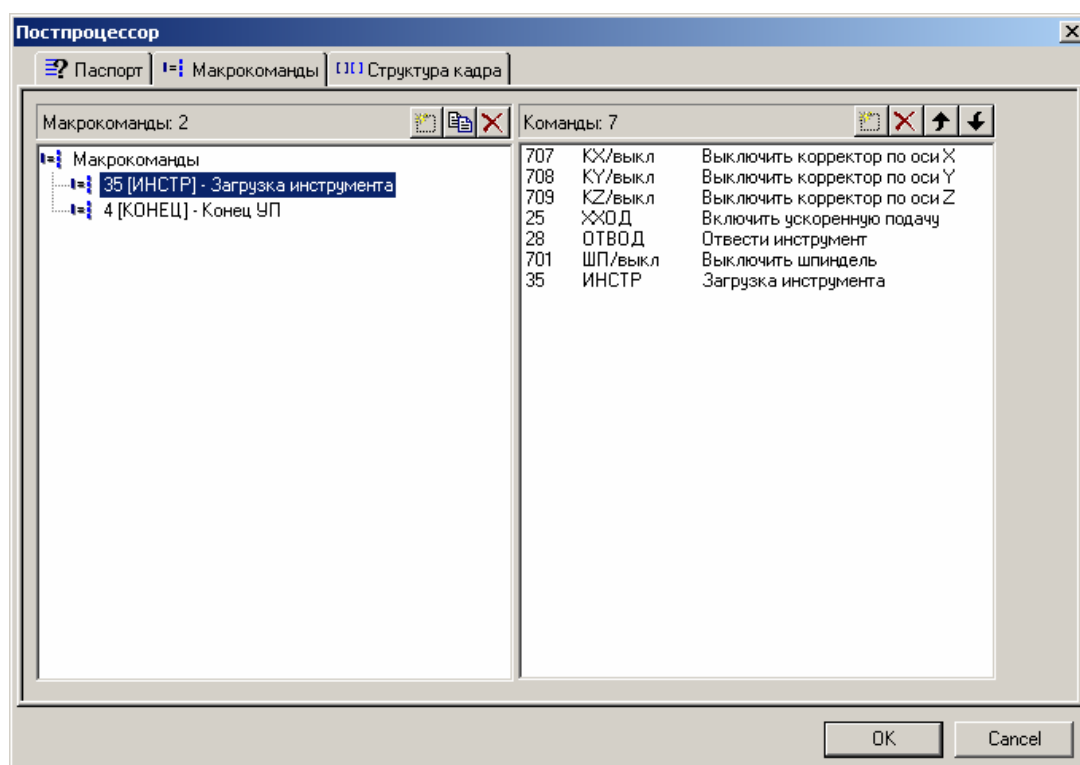


Рис. 12. Окна макрокоманд **Загрузка инструмента** и **Конец УП**.

Глава 5. Формирование макета кадра

Напоминаем, что *макет кадра* – это структура кадра управляющей программы: взаимное расположение всех возможных окон кадра и описание каждого из них.

Окно кадра описывает слово кадра управляющей программы и состоит из двух частей:

Символьная часть – адрес ЧПУ (может содержать несколько символов),
Формат вывода определяет вид выводимой числовой информации.

Пример: Окно кадра:

G[]

G - символьная часть окна,

[] - условное обозначение формата вывода.

Макет кадра:

N[]G[]G[]X[]Y[]Z[]I[]J[]K[]F[]M[]L[]

Файл макета кадра имеет имя, например, для постпроцессора с номером 222 - KADR0222.ANK. Без этого файла управляющая программа формироваться не будет, адаптер выдаст сообщение **"Нет макета кадра"**.

Пример файла макета кадра смотрите в [Приложении Г](#).

5.1. Формат вывода

Параметры формата вывода:

Тип окна

Определяет тип выводимой информации. Существуют следующие типы окон:

- *Номер кадра* используется для нумерации кадров управляющей программы.
- *Подготовительная функция*. Определяет режим и условия работы станка и УЧПУ. Например, включение линейной и круговой интерполяции. Параметры подготовительных функций определяются через другие слова кадра.
- *Вспомогательная функция*. Используется для включения вспомогательных функций станка, например включение/ выключение шпинделя.
- *Числовая информация* используется для вывода параметров подготовительных функций (например: координаты линейного и кругового перемещения, время выстоя) и формирования любых других функций, кроме нумерации кадров и формирования конца кадра.
- *Синхродорожка* используется для формирования пропусков в управляющей программе внутри кадра и между кадрами.
- *Текст* используется для формирования текстовой информации.
- *Символьное окно* используется для вывода только символьной части окна.
- *Конец кадра* используется для формирования конца кадра.

Литера окна

В данное окно вводится буква, обозначающая функцию.

Например, для функции линейной интерполяции G1 литерой окна будет G.

Интервал нумерации

В данную функцию вводится значение интервала нумерации кадров.

Например, для интервала нумерации 10 номера кадров будут идти:

N10

N20

N30

...

Номер группы окна

Номер группы окна (число от 0 до 49) определяет принадлежность окна к той или иной группе альтернативных функций станка. Например, альтернативными являются функции включения круговой и линейной интерполяции, то есть в одном кадре не может одновременно присутствовать G1 и G2 или G3.

Одной группе принадлежат также окна координат перемещений (X, Y и Z), так как литеры этих окон разные.

Признак модальности

Если определено, что окно действует модально, система будет запоминать последнее выведенное в это окно значение, и последующий вывод будет осуществляться только в том случае, если выводимое значение отличается от предыдущего.

Количество позиций

Максимальное количество символов выводимого в окно числа.

Например, если для окна G[] общее количество позиций равно 2, то в него можно вывести целые числа от -9 до 99. Все другие числа в заданное количество позиций не поместятся, адаптер выдаст соответствующее сообщение.

Вывод лидирующих нулей

Определяет, нужно ли выводить левые нули до достижения заданного максимального количества позиций.

Например, для окна G[] максимальное количество позиций равно 2 и в него нужно вывести число 1. На языке алгоритма такая команда может быть записана, как G->1;. Если для окна G[] определено, что левые нули нужно выводить, сформируется часть кадра:

G01

Если не нужно выводить:

G1

Количество позиций после десятичной точки

Определяет точность вывода числа.

Например, если указано количество позиций после десятичной точки 3, число при выводе в УП будет округляться до тысячных. (Округление производится по общим правилам: 0.5 округляется до 1.)

Вывод последних нулей

Определяет, нужно ли выводить незначащие нули до достижения заданного количества позиций после десятичной точки.

Например, для окна X[] определено, что количество позиций после десятичной точки равно 3 и выполняется команда X->12.5;.

Если для этого окна определено, что правые нули выводятся, сформируется слово:

X12.500

Если правые нули не выводятся:

X12.5

Вывод десятичной точки в целых числах

Определяет, нужно ли выводить десятичную точку в целых числах.

Например, выполняется команда X->12;.

Если для окна X[] определено, что десятичная точка должна быть выведена, сформируется часть кадра

X12.

Если определено, что десятичная точка не выводится:

X12

Вывод числа в виде целого количества дискрет

В этом случае число представляется в виде целого количества дискрет, имеющих фиксированную величину. Если число состоит не из целого количества дискрет, происходит округление выводимой величины.

Например, для окна X[] определен вывод в виде целого количества дискрет и выполняется команда X->12.5;. Если дискретность равна 0.01, после отработки команды сформируется часть кадра (12.5/0.01)

X1250

Если дискрета равна 0.005, то (12.5/0.005):

X2500

Вывод нулевых значений

Определяет, нужно ли выводить «0» в окно.

Например, выполнялась команда X->0;. Если для окна X[] определен вывод нулевой величины, сформируется часть кадра

X0

Если вывод нулевой величины не определен, сформируется часть кадра, содержащая только символьную часть окна:

X

Вывод знака "+" в положительных числах

Например, выполнялась команда X->12.5;.

Если определен вывод знака +, сформируется часть кадра

X+12.5

Если не определен:

X12.5

Ось действия окна

Определяет, по какой оси отрабатывается выводимая в окно величина. Используется только при формировании управляющей программы в

приращениях. В этом случае все погрешности, возникшие при округлении чисел, учитываются при выводе в это окно последующих значений, и все перемещения по этой оси автоматически сводятся в ноль.

5.2. Формирование окон различных типов

Запросы системы для различных типов окон:

НОМЕР КАДРА

- литера окна;
- количество позиций;
- интервал нумерации кадров;
- вывод лидирующих нулей.

ПОДГОТОВИТЕЛЬНАЯ и ВСПОМОГАТЕЛЬНАЯ ФУНКЦИИ

- литера окна;
- номер группы окна;
- количество позиций;
- вывод лидирующих нулей;.
- признак модальности.

ЧИСЛОВАЯ ИНФОРМАЦИЯ

- литера окна;
- номер группы окна;
- количество позиций;
- ось действия окна;
- подавление левых нулей;.
- признак модальности.
- вывод знака "+";

Остальные параметры зависят от того, в каком виде выводится число в кадр: в виде дроби или в виде целого количества дискрет.

При выборе в списке «Целое число дискрет»:

- величина дискретности;

При выборе в списке «Дробная часть»:

- вывод последних нулей;
- вывод нулевых значений;
- вывод десятичной точки;
- подавление правых нулей в дробных числах;
- ось действия окна.

СИНХРОДОРОЖКА

- литера окна.

ТЕКСТ

- литера окна.


СИМВОЛЬНОЕ ОКНО

- литера окна.

КОНЕЦ КАДРА

- литера окна.

5.3. Действия пользователя

Нажмите кнопку **Параметры**  Появится диалог «Постпроцессор». Выберите закладку **Структура кадра**.

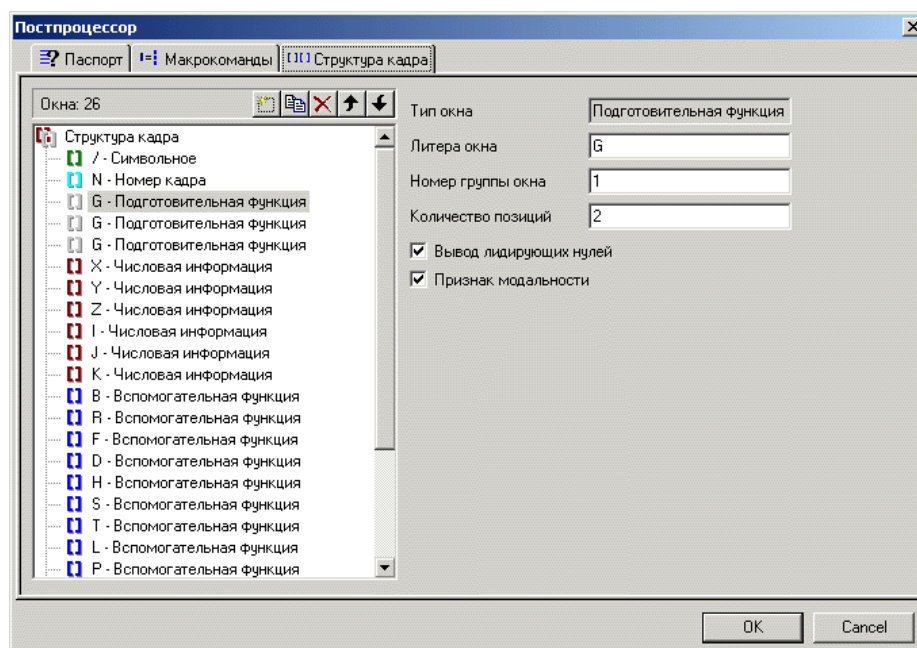



Рис. 13. Формирование макета кадра.


Формирование окна кадра происходит при помощи нажатия кнопки **Создать** . Откроется дополнительное меню. Необходимо выбрать тип окна и определить параметры формата вывода выбранного окна (смотрите разделы [5.1.](#) и [5.2.](#)).


ТИП ОКНА
НОМЕР КАДРА
ПОДГОТОВИТЕЛЬНАЯ ФУНКЦИЯ
ВСПОМОГАТЕЛЬНАЯ ФУНКЦИЯ
ЧИСЛОВАЯ ИНФОРМАЦИЯ
СИНХРОДОРОЖКА
ТЕКСТ
СИМВОЛЬНОЕ ОКНО
КОНЕЦ КАДРА

После этого можно описать новое окно, оно займет свое место после текущего и будет иметь очередной порядковый номер.

Для копирования удаления и перемещения используйте следующие кнопки:

Кнопка **Копировать**  копирует выделенный элемент.

Кнопка **Удалить**  удаляет выделенный элемент.

Кнопка **Переместить вверх**  перемещает вверх выделенный элемент вверх.

Кнопка **Переместить вниз**  перемещает вниз выделенный элемент вниз.

Глава 6. Формирование файла алгоритмов

Алгоритмы отображения действия команд CLDATA содержатся в файле алгоритмов. Этот файл является частью постпроцессора на станок и имеет имя, например, для анкеты с номером 222 - FTRP0222.ANK. Без этого файла управляющая программа формироваться не будет, адаптер выдаст сообщение **“Нет файла алгоритмического заполнителя”**.

Алгоритм представляет собой последовательность строк следующего формата:

[<метка>:] [ELSE] [IF <условие выполнения>] <команда алгоритма>;	
IF	Указывает, что команда должна быть выполнена только при соблюдении условия, идущего IF. Только строки с IF могут иметь альтернативные строки.
ELSE	Указывает, что данная строка является альтернативной.
<метка>	идентификатор строки при ссылках (целое положительное число).
<команда алгоритма>	определяет действие по формированию управляющей программы или изменению значения системной или пользовательской переменной.

Пример файла алгоритмов смотрите в [Приложении Д](#).

6.1. Арифметические действия и функции в алгоритмах

При составлении выражений допускаются следующие действия:

+	Сложение,
-	Вычитание,
*	Умножение,
:	Деление,
**	Возведение в степень,
&	Логическое умножение (и),
	Логическое сложение (или),
<	Меньше,
>	Больше,
<=	Меньше и равно,
>=	Больше и равно,
=	Равенство,
!=	Неравенство.

В алгоритмах могут использоваться следующие функции:

SQRT	Квадратный корень,
------	--------------------

SIN	Синус угла,
COS	Косинус угла,
TAN	Тангенс угла,
CTAN	Котангенс угла,
ASIN	Арксинус угла,
ACOS	Арккосинус угла,
ATAN	Арктангенс угла,
ACTN	Арккотангенс угла,
ABS	Абсолютная величина,
EXP	Экспонента,
LOGD	Десятичный логарифм,
LOGE	Натуральный логарифм,
COKN	Величина, выведенная в окно,
NTC	Номер точки цикла в таблице,
NNC	Номер начала цикла в таблице,
DEV	Вычисление целой части с округлением.
BMES	Сообщение, выдаваемое на экран. Имеет две кнопки «Ок» и «Cancel». Выход функции: 1 – Ok, 2 – Cancel.

Пример:

I=BMES('Продолжить?');

IF I=2 DO;

I=BMES('Формирование УП было прервано пользователем');

EXIT;

ENDDO;

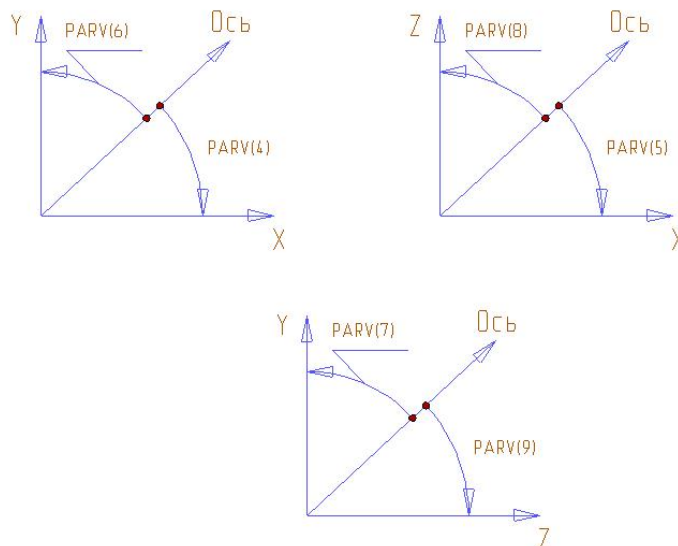
Появится сообщение «Продолжить?», если нажать «Cancel», то появится сообщение «Формирование УП было прервано пользователем» и формирование кадров УП прекратится.

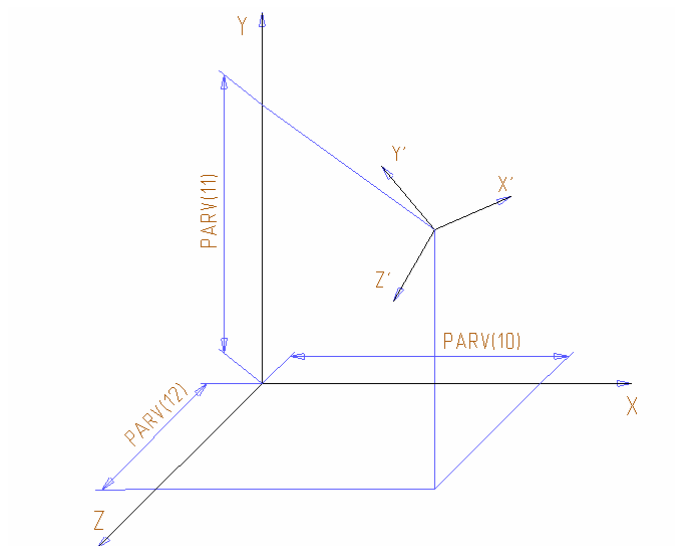
Примечание: Запрещается использовать в тексте сообщения символы @ и #.

PARV1	Направляющий косинус выбранной оси текущей системы координат к оси X базовой системы координат
PARV2	Направляющий косинус выбранной оси текущей системы координат к оси Y базовой системы координат
PARV3	Направляющий косинус выбранной оси текущей системы координат к оси Z базовой системы координат
PARV4	Угол, на который нужно повернуть текущую систему координат для того, чтобы проекция выбранной оси текущей

системы координат на плоскость XY базовой системы координат совместилась с осью X базовой системы координат.

- PARV5 Угол, на который нужно повернуть текущую систему координат для того, чтобы проекция выбранной оси текущей системы координат на плоскость XZ базовой системы координат совместилась с осью X базовой системы координат.
- PARV6 Угол, на который нужно повернуть текущую систему координат для того, чтобы проекция выбранной оси текущей системы координат на плоскость XY базовой системы координат совместилась с осью Y базовой системы координат.
- PARV7 Угол, на который нужно повернуть текущую систему координат для того, чтобы проекция выбранной оси текущей системы координат на плоскость ZY базовой системы координат совместилась с осью Y базовой системы координат.
- PARV8 Угол, на который нужно повернуть текущую систему координат для того, чтобы проекция выбранной оси текущей системы координат на плоскость XZ базовой системы координат совместилась с осью Z базовой системы координат.
- PARV9 Угол, на который нужно повернуть текущую систему координат для того, чтобы проекция выбранной оси текущей системы координат на плоскость ZY базовой системы координат совместилась с осью Z базовой системы координат.
- PARV10 Смещение нуля текущей системы координат по оси X базовой системы координат
- PARV11 Смещение нуля текущей системы координат по оси Y базовой системы координат
- PARV12 Смещение нуля текущей системы координат по оси Z базовой системы координат





XYZ – базовая система координат.

X'Y'Z' – текущая система координат.

Для определения приоритета выполнения действий используются скобки.

6.2. Команды алгоритмов

Присвоение значения

Команда изменяет значение системной или пользовательской переменной.

Пример:

```
_P1=10;
_P2=20;
_P3=_P1+_P2+15;
```

После отработки этих команд пользовательские переменные _P1, _P2 и _P3 примут значения 10, 20 и 45.

Занесение информации в окно макета кадра

Команда имеет следующий формат:

<номер окна в макете кадра>—><информация>;

Как видите, для вывода информации в окно необходимо указать его порядковый номер в макете кадра.

Пример: Инструмент находится в точке с координатами X=10, Y=20, Z=30.

Макет кадра:

N[G[G[X[Y[Z[M]....

Отработаны команды алгоритма:

```
4->XT;
5->YT;
6->ZT;
```


2→1;

Сформировалась часть кадра:

G1X10.Y20.Z30.

Если в алгоритмах номер окна указывать числом (как в предыдущем примере), при корректировке макета кадра может возникнуть необходимость редактирования всех алгоритмов, так как может произойти смещение окон. Чтобы избежать этого, номера окон рекомендуется присваивать пользовательским переменным в алгоритме команды *Программа* (код 1). Она всегда будет первой в CLDATA, соответственно ее алгоритм обрабатывается всегда первым.

Пример Инструмент находится в точке с координатами X=10, Y=20, Z=30.

: Макет кадра:

N[]G[]G[]X[]Y[]Z[]M[]....

Алгоритм команды *Программа*:

_G=2 ;

_G2=3 ;

_X=4 ;

_Y=5 ;

_Z=6 ;

Отработаны команды алгоритма:

_X→XT ;

_Y→YT ;

_Z→ZT ;

_G→1 ;

Сформировалась часть кадра:

G1X10.Y20.Z30.

В этом случае при корректировке макета кадра будет достаточно исправить только алгоритм команды *Программа*.

АОКНО – Вывод в УП символьной части окна

АОКНО <номер окна>;

Пример: Макет кадра:

N[]/REM_TOOL[]G[]G[]X[]Y[]Z[]M[]....

Отработанная команда:

АОКНО 2;

Сформированная часть кадра:

/REM_TOOL

GOTO – переход на метку

Команда передачи управления на метку.

GOTO <номер метки>;

Пример: 1: _P2=1;

.

.

IF _P2=0 GOTO 1;

FOR <инициализация переменной> <условие выполнения>
<выражение итерации> – *цикл FOR*

...

ENDDO;

Пример:

FOR _I=1 _I<100 _I=_I+1;

...

ENDDO;

КАДР – *Формирование конца кадра*

Автоматически формируется номер кадра и конец кадра. Если в окнах не было выведенной информации, эта команда игнорируется.

КАДР;

Пример:

_G->1;

КАДР;

_G->2;

КАДР;

КАДР;

сформируются кадры:

N001G01

N002G02

обратите внимание на то, что последняя команда КАДР не выполнялась.

ПНКАДР – *Подавление номера кадра*

Команда отменяет автоматическое формирование номера в текущем кадре. Порядок нумерации сохраняется.

ПНКАДР;

Пример:

_G->10;

КАДР;

_G->20;

ПНКАДР;

КАДР;

_G->30;

КАДР;

сформируются кадры:

N001G10

G20

N003G30

ИНКАДР – *Инициализация номера кадра*

Команда устанавливает значение, с которого пойдет дальнейшая нумерация кадров.

ИНКАДР <число>;

Пример _G->10;
: КАДР;
 _G->20;
 ИНКАДР 100;
 КАДР;
 _G->30;
 КАДР;

Сформируются кадры:

N001G10
N101G20
N102G30

MODOFF – Подавление модальности окна

Игнорирование признака модальности окна в текущем кадре. После вывода кадра признак модальности этого окна восстанавливается.

Команда имеет следующий формат:

MODOFF <номер окна>;

Пример: _G->1;
 _X->100;
 КАДР;
 _G->1;
 _Y->200;
 КАДР;
 _G->1;
 _X->300;
 MODOFF _G;
 КАДР;
 _G->1;
 _Y->400;
 КАДР;

Сформируются кадры:

N001G01X100
N002Y200
N003G01X300
N004Y400

CALL – Вызов алгоритма

Команда используется для выполнения какого-либо алгоритма в качестве подпрограммы. Номером алгоритма считается код команды CLDATA или число, которое вы определили вместо кода команды CLDATA при составлении алгоритма.

CALL <номер алгоритма>;

Пример: Предположим, обрабатывается следующий алгоритм:

_G->1;
_X->100;
CALL 600;
КАДР;

Алгоритм с номером 600 выглядит следующим образом:

_L->801;

Сформируется кадр:
N001G01X100L801

DO, ENDDO – Операторные скобки

Команда используется для выделения строк алгоритма, которые должны быть выполнены при общем условии. Каждая открывающая операторная скобка DO должна закрываться командой ENDDO.

```
DO;  
...  
ENDDO;
```

Пример: Предположим, обрабатывается следующий алгоритм:

```
IF _P1=1 DO;  
  _X->100;  
  _Y->100;  
ENDDO;  
ELSE DO;  
  _X->200;  
  _Y->200;  
ENDDO;  
_G->1;  
КАДР;
```

Если _P1=1, сформируется кадр

N001G01X100Y100

Если нет, то

N001G01X200Y200

BEGIN – Начать преобразование команд CL Data в кадры УП заново.

Команда используется в случае, когда необходимо считать какие-то данные из файла CL Data. Например, может возникнуть необходимость в начале УП перечислить все точки, на которых будут обрабатываться стандартные сверлильно-расточные циклы. По этой команде обработка текущей команды CL Data останавливается, и преобразование начинается заново, с первого алгоритма.

Примечание:

- Сформированные к этому моменту кадры УП не уничтожаются!
- Адаптер не ограничивает количество таких преобразований, поэтому избегайте закливания!

```
BEGIN;
```

Пример: Предположим, обрабатывается следующий алгоритм:

```
IF _P1=0 DO;  
  _P1=1; - меняем значение пользовательской переменной, что  
  бы при следующей трансляции CL Data не произошло  
  закливания в этом алгоритме.  
BEGIN; - начинаем заново формировать кадры УП.
```

ENDDO;

EXIT

Прекращение формирования управляющей программы.

BMES ('текст сообщения')

Выдача сообщений во время работы адаптера.

Пример: Предположим, нужно вывести сообщение «Продолжить формирование УП?»:

_MES=BMES('Продолжить формирование УП?'); -
появится меню с текстом «Продолжить формирование
УП?» и двумя кнопками «Ок» и «Cancel».
IF _MES=2 EXIT; - если нажали на кнопку «Cancel»,
прекратить формирование управляющей программы

Переменная _MES может принимать следующие
значения:

- 1 – нажатие на кнопку «Ок»
- 2 - нажатие на кнопку «Cancel»

ПОДВОД

Формировать параметры подвода. Формат команды: ПОДВОД <ось>
<расстояние>. Команда задает расстояние до точки переключения на рабочую
подачу, в которой должны быть включены шпиндель, СОЖ, и ось, по которой
должно быть выдержано это расстояние. После подвода инструмента будет
включен алгоритм с номером 712, если он есть, затем будут сгенерированы
команды на включение шпинделя и СОЖ, если они были заданы. Команда
ПОДВОД действует модально по всем объектам. Пока адаптер выполняет
действия, связанные с командой ПОДВОД, системная переменная НПОДВОД
(BEGSTART) имеет значение ВКЛ, в остальных случаях – ВЫКЛ. Пример:

В файле алгоритмов задано: ПОДВОД Z 50;

Фрагмент CLDATA имеет следующий вид:

НЦ/0 0 100;
ИНСТР/...
ШПИНД/..
СОЖ/...
ХХОД;
ИДИТОЧ/10 15 100;
ИДИТОЧ/10 15 2;
ПОДАЧА/20 мм/мин;
ИДИТОЧ/10 15 -8;

.
.

Адаптер сгенерирует следующую последовательность команд:

401 (НЦ/0 0 100)

35 (ИНСТР/...)

25 (ХХОД)

181 (ИДИТОЧ/10 15 100)

181 (ИДИТОЧ/10 15 52)

712 (Если она присутствует в файле алгоритмов)

24 (ШПИНД/..)

26 (СОЖ/...)

181 (ИДИТОЧ/10 15 2)

23 (ПОДАЧА/20 мм/мин)

181 (ИДИТОЧ/10 15 -8)

.

.

CIRC/ON

Включить круговой интерполятор.

CIRC/OFF

Выключить круговой интерполятор.

СКТЦ

RPCYC

Считать координаты точки цикла из таблицы. Формат команды: СКТЦ <порядковый номер точки в таблице>. После выполнения команды в системные переменные ХТТЦ(Х/ТРС), УТТЦ(У/ТРС) и ЗТТЦ(З/ТРС) запишутся соответственно координаты Х, У и Z точки из таблицы. Используется для создания таблицы координат точек стандартных циклов в управляющей программе перед обработкой (в кадрах обработки используются не координаты точек, а номера точек в таблице).

СТИНСТ

RTOOLPAR

Считать параметры инструмента из таблицы. Формат команды: СТИНСТ <порядковый номер инструмента в таблице>. После выполнения команды в системные переменные НПИТ(NPT/TBL), НКХТ(ХСТ/TBL), НКУТ(УСТ/TBL) и НКЗТ(ЗСТ/TBL) запишутся соответственно номер позиции, номера корректоров по осям Х, У и Z соответственно. Используется для создания таблицы параметров инструмента в управляющей программе перед обработкой (в кадрах обработки используются не параметры инструмента, а номера записей в таблице).

СТНЦ

RHOME

Считать параметры команды НЦ из таблицы. Формат команды: СТНЦ <порядковый номер НЦ в таблице>.. После выполнения команды в системные переменные ХТТЦ(Х/ТРС), УТТЦ(У/ТРС) и ЗТТЦ(З/ТРС) запишутся

соответственно координаты X, Y и Z точки из таблицы. Используется для создания таблицы параметров НЦ в управляющей программе перед обработкой (в кадрах обработки используются не параметры команды НЦ, а номера записей таблице).

СКТЦТ RQPCYC

Считать количество точек, на которых выполняется текущий цикл. Формат команды: СКТЦТ <порядковый номер цикла>. Система формирует команды сверлильно–расточных циклов в следующей последовательности:

.
Другая команда/
ИДИ/ТОЧ... (код 181)
ЦИКЛ/... (код 36) // Цикл первый в группе (номер п.п. 1)
ИДИ/ТОЧ... (код 181)
ЦИКЛ/... (код 36) // Цикл не первый в группе (номер п.п. 2)
ИДИ/ТОЧ... (код 181)
ЦИКЛ/... (код 36) // Цикл не первый в группе (номер п.п. 3)
Другая команда/

.
ИДИ/ТОЧ... (код 181)
ЦИКЛ/... (код 36) // Цикл первый в группе (номер п.п. 4)
ИДИ/ТОЧ... (код 181)
ЦИКЛ/... (код 36) // Цикл не первый в группе (номер п.п. 5)
Другая команда/

.
Адаптер сортирует все вызовы циклов по группам и запоминает порядковые номера циклов первых в группе. Этим номерам ставится в соответствие количество точек, на которых выполняется группа циклов. Всем остальным номерам ставится в соответствие ноль. Если <порядковый номер цикла> соответствует первому в группе циклов, то после выполнения команды СКТЦТ <порядковый номер цикла>; в системную переменную КТЦТ(NOF/INC) запишется количество точек, на которых выполняется группа циклов. Если <порядковый номер цикла> не соответствует первому в группе циклов, то после выполнения команды СКТЦТ <порядковый номер цикла>; в системную переменную КТЦТ(NOF/INC) запишется ноль. В приведенном примере после выполнения команды СКТЦТ 1; системная переменная КТЦТ будет иметь значение 3, после выполнения команды СКТЦТ 4; системная переменная КТЦТ будет иметь значение 2, после выполнения команд СКТЦТ 2; СКТЦТ 3; СКТЦТ 5; системная переменная КТЦТ будет иметь значение 0. Эту команду удобно использовать, когда в управляющей программе после объявления цикла идет перечисление номеров точек выполнения цикла согласно таблице. По переменной КТЦТ определяется цикл, первый в группе, затем через функцию NTC определяются все номера точек выполнения согласно таблице.

//

Комментарий внутри алгоритма. После // обязателен пробел, за ним текст комментария, заканчивающийся символом «;». Используется для пояснений внутри теста алгоритма.

INT/ON

Включить интервал между словами кадра. После выполнения этой команды все слова в кадре будут отделены пробелами.

GNBL/ON

Включить нумерацию кадров. После выполнения этой команды все кадры будут нумероваться.

GNBL/OFF

Выключить нумерацию кадров. После выполнения этой команды кадры не будут нумероваться.

SET/FILE

Установить имя файла, в который будут выводиться кадры управляющей программы. По умолчанию устанавливается файл PLENT.TAP. Если формируется одновременно несколько файлов, переключение между ними осуществляется этой командой. При этом если в указанный файл были сделаны записи, все последующие будут добавлены. Все файлы формируются в рабочей директории, полный путь задавать недопустимо.

Формат команды SET/FILE 'имя файла с расширением'.

COPYFILE

Скопировать файл управляющей программы.

COPYFILE ' имя файла с расширением '.

SETEL

Установить элемент контура.

SETEL < номер элемента >

NCD

Поиск фразы с заданным кодом от текущего положения до смены инструмента или до конца программы.

NCD < код фразы CLData >

DATAV

Вычислить параметры оси A, B или C текущей системы координат относительно базовой.

DATAV {A, B или C} TR1 TR2;

TR1 – текущая система координат; TR2 – базовая система координат.

ROTECOOR

Повернуть текущую систему координат вокруг оси.
ROTECOOR {A, B или C} TR1 V;
TR1 – текущая система координат; V – угол поворота в радианах.

CLCOOR

Вычислить положение текущей системы координат относительно базовой.

CLCOOR TR1 TR2

TR1 – текущая система координат; TR2 – базовая система координат.

GETCODE(<номер фразы CLData>) - установить код фразы CLData.

GETLONG(<номер фразы CLData>) - установить длину фразы CLData.

GETQC(<код>) - установить количество фраз CLData в основной программе или подпрограмме. 0 – основная программа, 1 – подпрограмма.

Пример: _QC=GETQC(0); - в пользовательскую переменную _QC запишется количество фраз CLData в основной программе.

GETFR <номер фразы CLData> _P1[] - прочитать фразу CLData в массив. _P1 – имя массива.

SETPROG - установить на обработку файл FCLD, содержащий основную программу.

SETSUB - установить на обработку файл CLDP, содержащий подпрограмму.

SETWRF <‘имя файла’> - установить файл записи с указанным именем.

SETPF <‘имя файла’> - установить файл записи как CLData основной программы.

SETSF <‘имя файла’> - установить файл записи как CLData подпрограммы.

WRCOM _P1[] – сформировать команду CLData, используя массив данных с именем _P1.

EALG – вызывать текущий алгоритм перед трансляцией каждой команды CLData.

Пример:

```
500;  
EALG;  
BLOCK;  
_TXT->'НОВАЯ КОМАНДА CLDATA';  
BLOCK;  
END;
```

Теперь перед тем как обработать новую команду CLData, адаптер выполнит 500 алгоритм. В тексте УП это будет выглядеть так:

```
N100 G1 X10 Y20  
N110 НОВАЯ КОМАНДА CLDATA.  
N100 G1 X150 Y25
```

<имя массива>[]=<значение> - инициализация массива элементов. Все элементы массива с указанным именем примут указанное значение.

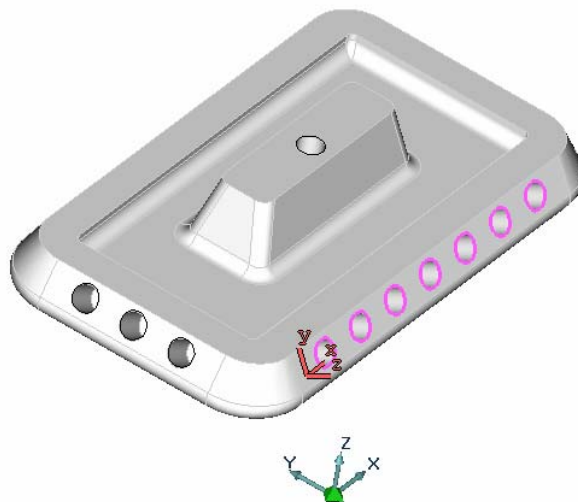
Пример: `_P1[]=10;` - инициализируется массив с именем `_P1`, все элементы которого равны 10.

<имя переменной>=<имя массива>[номер элемента массива] – присваивание переменной значения элемента массива.

Пример: `_NAME=_P1[2];` - переменной `_NAME` присваивается значение 2 элемента массива `_P1`.

6.3. Пример работы с трансформами.

Рассмотрим способы работы с трансформами на примере 5-осевого сверления без смены системы координат детали.



Расчетная (базовая) система координат -

Повернутая система координат -

система координат станка, на рисунке ее оси показаны зеленым цветом.

система координат плоскости, в которой расположены отверстия, на рисунке ее оси показаны красным цветом.

Как видно из рисунка обрабатываемые отверстия (подсвечены малиновым цветом) находятся в плоскости, расположенной под определенным углом к расчетной системе координат. При расчете CL Data ADEM CAM формирует:

1. трансформ для пересчета координат перемещений инструмента из повернутой системы координат в базовую.
2. координаты перемещений инструмента относительно повернутой системы координат.

В тексте CL Data это отражается как «**Phraze Code 10123**». 10123 – это номер команды CL Data. При формировании кадров УП, адаптер автоматически пересчитывает с помощью сформированного трансформации координаты перемещений относительно расчетной системы координат. Далее приведен текст алгоритма №10123, в котором формируются данные для поворота оси инструмента на нужный угол.

10123;

DATAV C CLDCOOR SYSCOOR;

- вычисляем положение оси C (в нашем случае оси Z) относительно базовой системы координат.

_ARAD=PARV(1);

- присвоение пользовательской переменной _ARAD значения направляющего косинуса выбранной оси текущей (повернутой) системы координат к оси X базовой системы координат

_BRAD=PARV(2);

- присвоение пользовательской переменной _BRAD значения направляющего косинуса выбранной оси текущей (повернутой) системы координат к оси Y базовой системы координат

_CRAD=PARV(3);

- присвоение пользовательской переменной _CRAD значения направляющего косинуса выбранной оси текущей (повернутой) системы координат к оси Z базовой системы координат

CALL 40;

- вызов алгоритма №40 (поворот) для формирования поворота оси инструмента.

END;

- конец алгоритма

6.4. Примеры работы с пользовательскими командами и циклами

В системе ADEM имеется возможность в дополнение к существующим средствам проектирования маршрута обработки (технологические команды, технологические переходы и конструктивные элементы) создавать свои собственные команды и циклы обработки. Пользовательские команды и циклы создаются на основе так называемых «ini-файлов». Эти файлы находятся в папке ...ADEM/NCM/NCALG/INI и имеют формат <имя файла>.ini.

- Примечание:**
- Изменять параметры Вы можете только у файлов, чье имя содержит слово «**user**» или «**mycycle**». Далее идет перечень ini-файлов, поставляемых в стандартной поставке ADEM.
 - boreuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Расточить».
 - centuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Центровать».
 - clearpuser.ini – настройка параметров закладки «Параметры пользователя» в технологической команде «Плоскость холостых ходов».
 - cutuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Резать».
 - drilluser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Сверлить».
 - enlarguser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Зенкеровать».
 - homeuser.ini – настройка параметров закладки «Параметры пользователя» в технологической команде «Начало цикла».
 - lboreuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Расточить (токарный)».
 - lcutuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Отрезать».
 - lfaceuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Подрезать».
 - lthreaduser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Нарезать резьбу (токарный)».
 - milluser.ini – настройка параметров закладки «Параметры пользователя» во фрезерных технологических переходах.

- punchuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Пробить».
- reamuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Развернуть».
- reclawuser.ini – настройка параметров закладки «Параметры пользователя» в технологической команде «Перезахват».
- rotationuser.ini – настройка параметров закладки «Параметры пользователя» в технологической команде «Поворот».
- safeposuser.ini – настройка параметров закладки «Параметры пользователя» в технологической команде «Безопасная позиция».
- tapuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Нарезать резьбу (фрезерный)».
- tooluser.ini – настройка параметров закладки «Параметры пользователя» в технологической команде «Инструмент»
- turnuser.ini – настройка параметров закладки «Параметры пользователя» в технологическом переходе «Точить».
- user0045.ini – пример для составления пользовательских команд.
- mусycle.ini – пример для составления пользовательских циклов.

6.4.1. Пример работы с пользовательскими командами


Рассмотрим приемы работы с пользовательскими командами на следующем примере: создадим на основе пользовательской команды № 45, содержащейся в стандартной поставке ADEM CAM, новую команду № 46 «Определение заготовки».



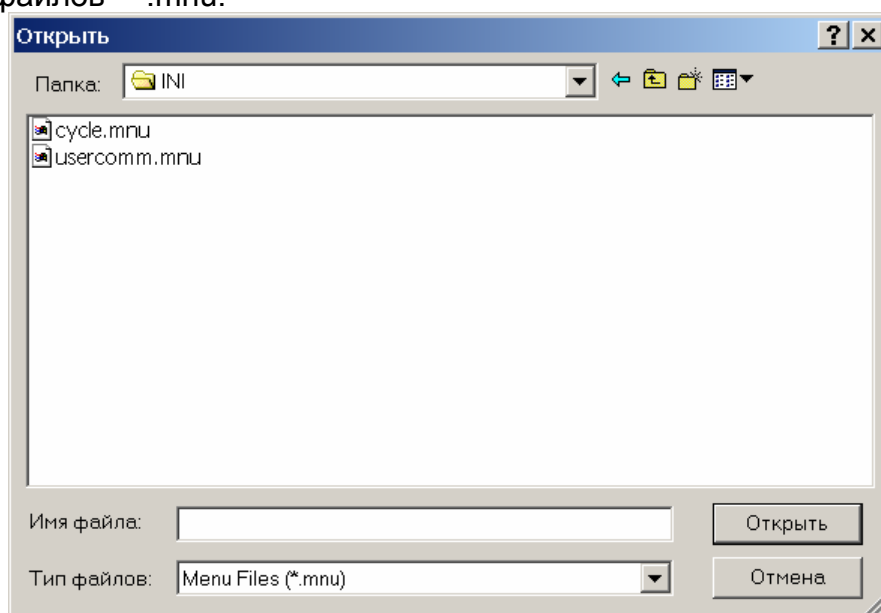
Примечание

Пользовательским командам можно присваивать любой номер от 0 до 9999.

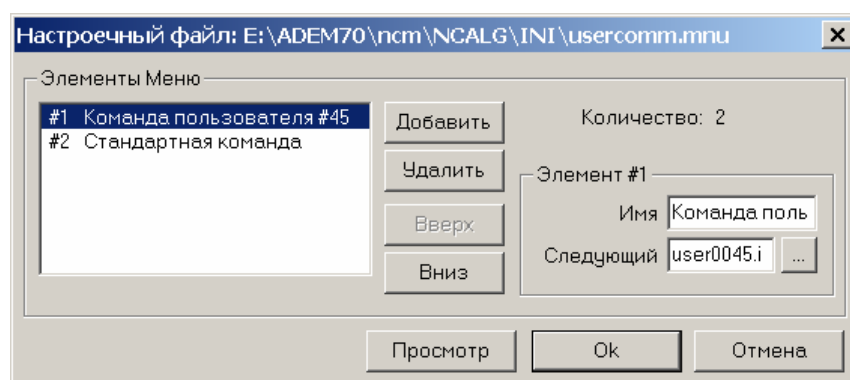
Для того чтобы сформировать или отредактировать команду пользователя Вам необходимо выполнить следующие действия:

1. Нажмите кнопку , расположенную на панели инструментов.

2. В появившемся диалоге откройте папку INI. В поле «Тип файлов» выберите тип файлов - *.mnu.



3. Откройте настроечный файл usercomm.mnu.



В состав диалога входят:

Элементы Меню:

Список	В окне отображается список элемент меню, состоящий из номера элемент меню и его имени. Чтобы установится на требуемый элемент меню, необходимо отметить его в списке.
Добавить	Добавить новый элемент меню в список. Элемент добавляется в конец списка.
Удалить	Удалить отмеченный элемент меню из списка.
Вверх	Переместить отмеченный элемент меню вверх списка.
Вниз	Переместить отмеченный элемент меню вниз списка.
Просмотр	используется для предварительного просмотра полученного меню.

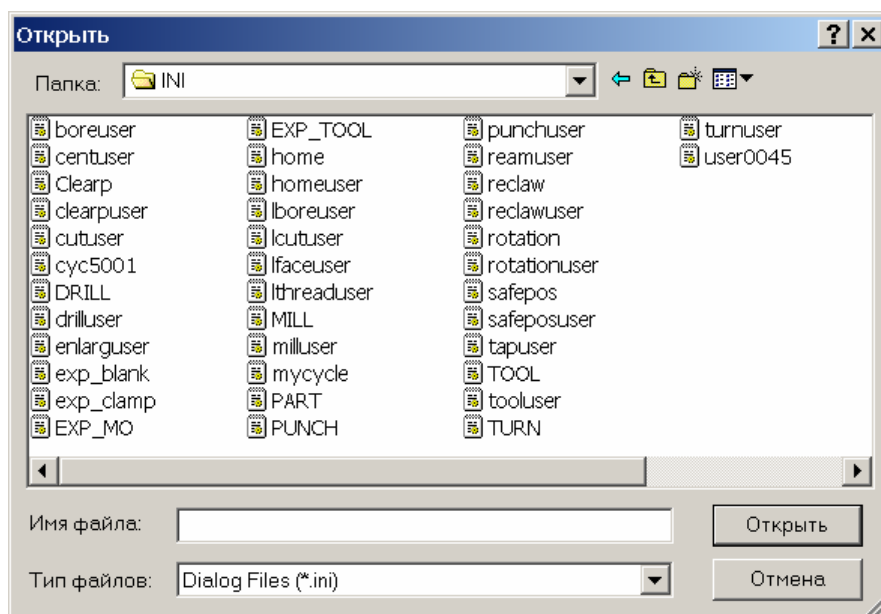
Элемент #1:

Имя	Имя элемента меню.
------------	--------------------

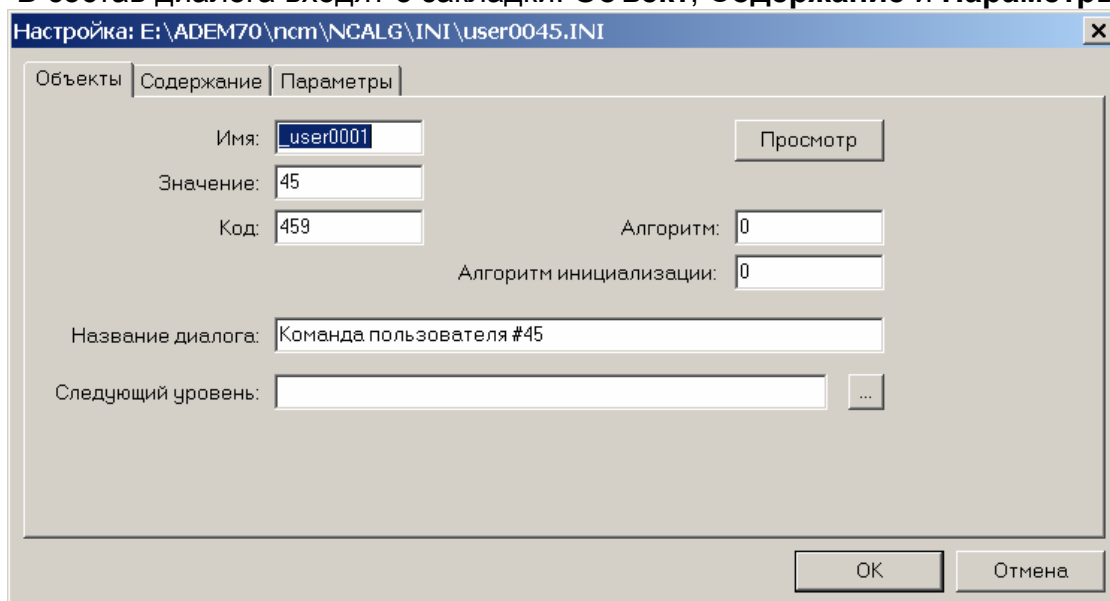
Следующий

Имя настроечного файла (*.ini) или файла меню (*.mnu). Устанавливает последовательность действий, которые будут произведены пользователем при выборе элемент меню. Если установлено имя настроечного файла, откроется диалог “Параметры Объекта”, который формируется по шаблону данного настроечного файла. Если установлено имя файла меню, откроется следующее меню выбора, созданное на основе данного файла меню.

4. Добавьте новую команду с именем «Создание заготовки», в качестве следующего укажите файл **user0046.INI**.
5. Нажмите **OK** и сохраните настроечный файл.
6. Повторите шаги 1 и 2, только теперь выберите тип файлов - *.ini, и откройте файл **user0045.INI**.



7. В состав диалога входят 3 закладки: **Объект**, **Содержание** и **Параметры**.



Закладка «Объект»:

Имя	Имя переменной, которая соответствует текущему объекту. Используется в алгоритмах.
Значение	Значение, которым инициализируется переменная, описанная параметром Имя (для пользовательских команд – это номер команды).
Код	Код объекта. Используется в алгоритмах для сортировки объектов по уровням (для пользовательских команд всегда 459).

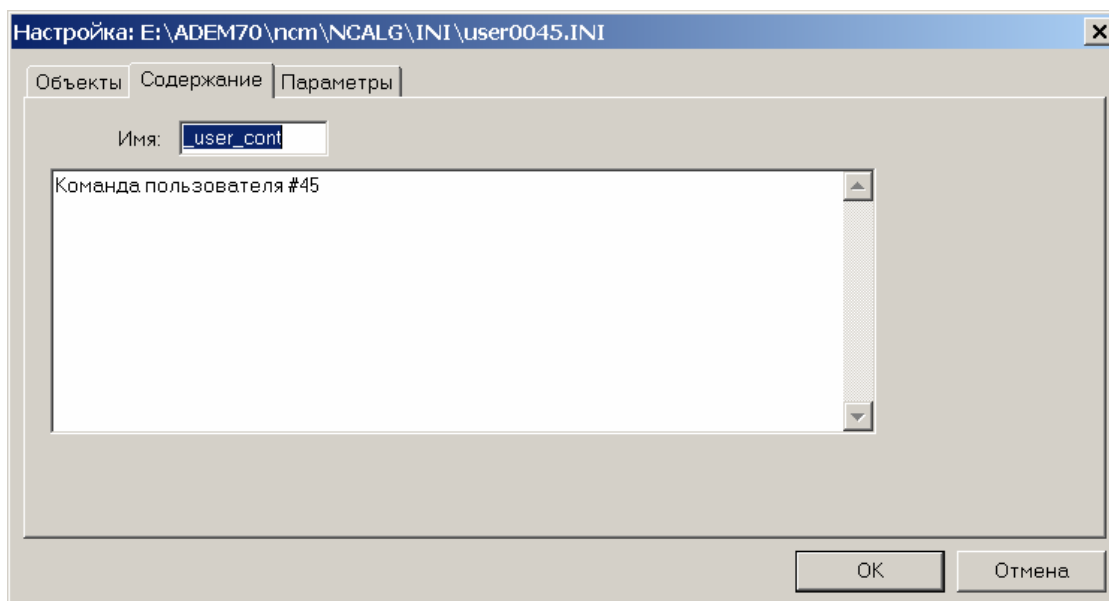


Примечание

Объекты с кодом 99 имеют другой вид. В отличие от обычных объектов с кнопками «ОК» и «Отмена», у него только кнопка «ЗАКРЫТЬ»

Алгоритм	Номер алгоритма. Если установлено не нулевое значение, активизируется кнопка “Алгоритм”, при нажатии на которую выполняется требуемый алгоритм. Содержится алгоритм в файле с именем: 0000<номер алгоритма>.alg.
Алгоритм инициализации	Номер алгоритма инициализации. Если установлено не нулевое значение, при создании объекта выполняется требуемый алгоритм, который инициализирует параметры диалога. Содержится алгоритм в файле с именем: 0000<номер алгоритма>.alg.
Название диалога	Заголовок диалога.
Следующий уровень	Имя настроечного файла (*.ini) или файла меню (*.mnu). Устанавливает последовательность действий, которые будут произведены пользователем при создании объектов на следующем уровне. Если установлено имя настроечного файла, то при выполнении команды “Создать Объект” на следующем уровне откроется диалог “Параметры Объекта”, который формируется по шаблону данного настроечного файла. Если установлено имя файла меню, то при выполнении команды “Создать Объект” на следующем уровне откроется меню выбора, созданное на основе данного файла меню.
Просмотр	Используется для предварительного просмотра полученного диалога “Параметры Объекта”.

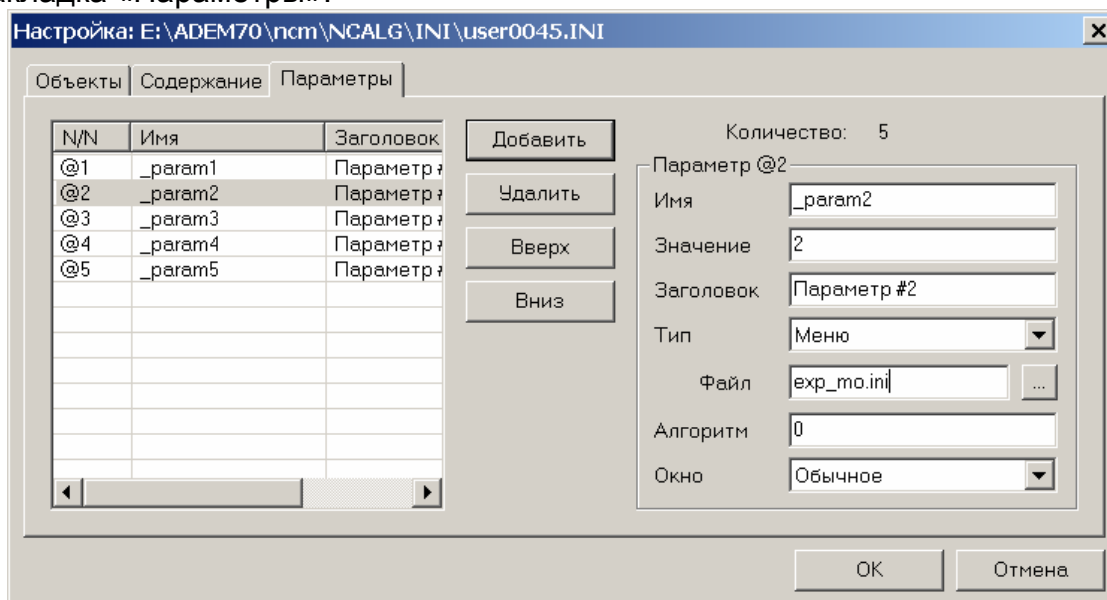
Закладка «Содержание»:



Имя Имя переменной содержания. Используется в алгоритмах.

Текст содержания Содержание используется для описания объекта. Текст является параметрическим. Связь с параметрами объекта устанавливается с помощью специального символа @, после которого устанавливается либо порядковый номер параметра, либо имя параметра, заключенное в [] скобки.

Закладка «Параметры»:



Список В окне отображается список параметров объекта, состоящий из номера параметра и имени его переменной. Чтобы установится на требуемый параметр, необходимо отметить его в списке.

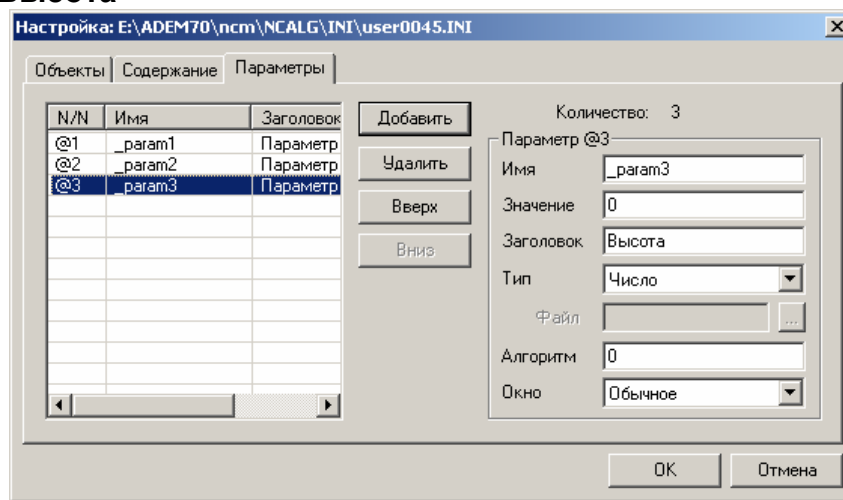
Добавить Добавить новый параметр в список. Параметр добавляется в конец списка.


Удалить Удалить отмеченный параметр из списка.

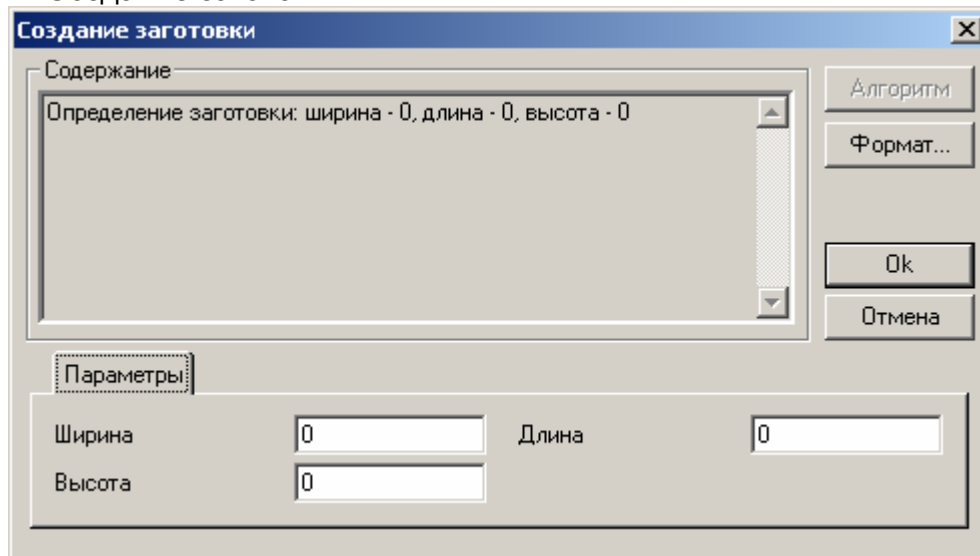
Вверх	Переместить отмеченный параметр вверх списка.
Вниз	Переместить отмеченный параметр вниз списка.
Поле «Параметр @2:	
Имя	Имя переменной текущего параметра. Используется в алгоритмах.
Значение	Значение, которым инициализируется переменная параметра.
Заголовок	Заголовок параметра.
Тип	<p>Тип параметра. Может принимать одно из следующих значений: <i>число</i>, <i>символ</i>, <i>дата</i>, <i>меню</i>, <i>меню2</i>, <i>вкладка</i>.</p> <ul style="list-style-type: none"> • <i>Число</i>. В качестве значения параметра могут использоваться математические выражения. Результат выполнения будет занесен в переменную параметра. • <i>Дата</i>. Отображается стандартный управляющий элемент диалога для ввода даты. • <i>Меню</i> и <i>меню2</i>. Комбинированный список, элементами которого являются строки текстового файла. Если определен тип <i>меню</i>, то после выбора строки в переменную параметра занесется число, соответствующее порядковому номеру строки в файле. Если определен тип <i>меню2</i>, то – сама строка. • <i>Вкладка</i>: Добавляет вкладку в диалог. В поле «Заголовок» необходимо ввести название вкладки. На вкладку помещаются все параметры находящиеся между двумя параметрами «Вкладка» или все параметры от текущего параметра «Вкладка» до конца. Если объект не имеет вкладок, то все элементы помещаются на вкладку – «Параметры».
Файл	Имя текстового файла, связанного с данным окном, тип которого установлен как <i>меню</i> или <i>меню2</i> .
Алгоритм	Номер алгоритма. Если установлено не нулевое значение, справа от значения появится кнопка “А”, при нажатии на которую выполняется требуемый алгоритм. Содержится алгоритм в файле с именем: <i>0000<номер алгоритма>.alg</i> .
Окно	Тип окна вывода параметра. Может принимать одно из 3-х значений: <i>обычное</i> , <i>большое</i> , <i>скрытое</i> , <i>обычное только чтение</i> , <i>большое только чтение</i> . Тип окна <i>большое</i> могут принимать только нечетные параметры: 1-й, 3-й... Если установлен тип <i>скрытое</i> , параметр не будет отображаться в диалоге и не может корректироваться пользователем. Параметры, имеющие тип окна <i>обычное только чтение</i> и <i>большое только чтение</i> могут изменять свои значения только из алгоритмов, в режиме редактирования их значения изменить нельзя. Данный вид параметра диалога

распространяются только на тип данных *число* и *строка*.
На другие типы данных параметр окна *только чтение* игнорируется.

8. На закладке «Объекты» в поле «Название диалога» напишите: **Определение заготовки**.
9. На закладке «Содержание» в поле «Текст содержания» напишите:
Определение заготовки: ширина - @[param1], длина - @[param2], высота - @[param3]
10. На закладке «Параметры» удалите все параметры и добавьте новые:
 - **Ширина**
 - **Длина**
 - **Высота**



11. Нажмите **OK** и сохраните настроечный файл под именем **user0046.ini**.
12. В модуле **ADEM CAM** нажмите кнопку  и выберите в контекстном меню пункт «Создание заготовки».



13. Введите параметры и нажмите **OK**. В маршруте обработки появится объект «**User Function: Создание заготовки**».
14. Для того, чтобы параметры команды пользователя отразились в кадрах УП, в постпроцессоре необходимо описать порядок действий в алгоритме № 459.
459;
IF N/USFUNC=46 DO;

```

_TXT->'ШИРИНА: @[PUSFUN1]';
BLOCK;
_TXT->'ДЛИНА: @[PUSFUN2]';
BLOCK;
_TXT->'ВЫСОТА: @[PUSFUN3]';
BLOCK;
ENDDO;
END;

```

После формирования в кадрах УП Вы увидите следующее:

```

...
N2 ШИРИНА: 10
N3 ДЛИНА: 50
N4 ВЫСОТА: 20
...

```

6.4.2. Пример работы с пользовательскими циклами


Рассмотрим приемы работы с пользовательскими циклами на следующем примере: создадим на основе цикла пользователя, содержащегося в стандартной поставке ADEM CAM, новый цикл № 10 «Расточка шпоночного паза».

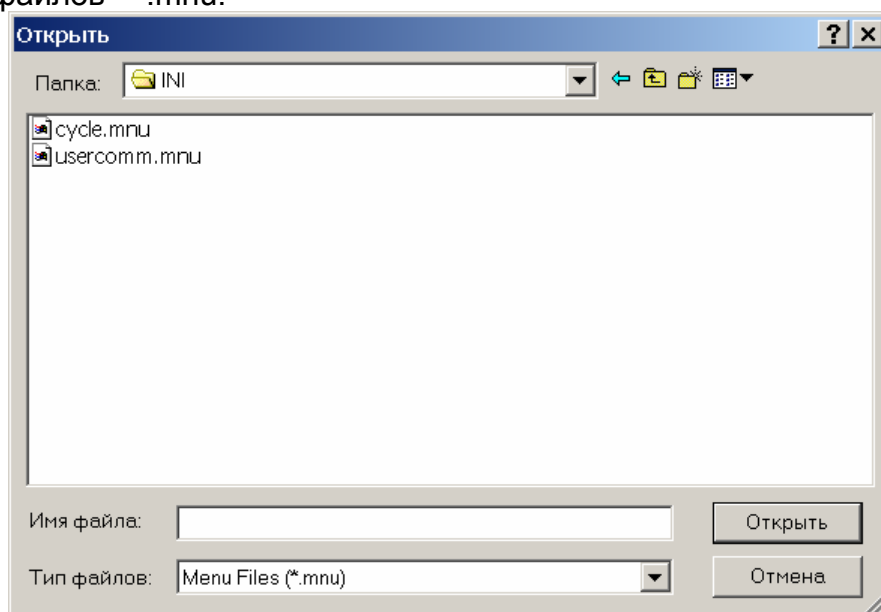


Примечание

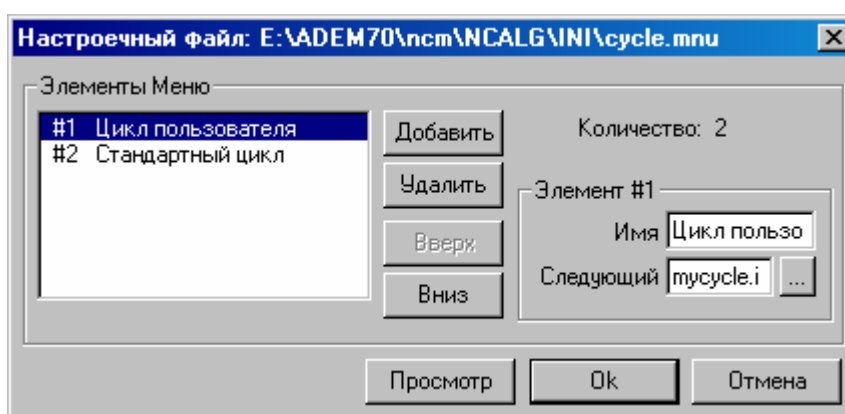
Пользовательским циклам можно присваивать любой номер от 0 до 9999.

Для того чтобы сформировать или отредактировать цикл пользователя Вам необходимо выполнить следующие действия:

1. Нажмите кнопку , расположенную на панели инструментов.
2. В появившемся диалоге откройте папку INI. В поле «Тип файлов» выберите тип файлов - *.mnu.



3. Откройте настроечный файл cycle.mnu.



В состав диалога входят:

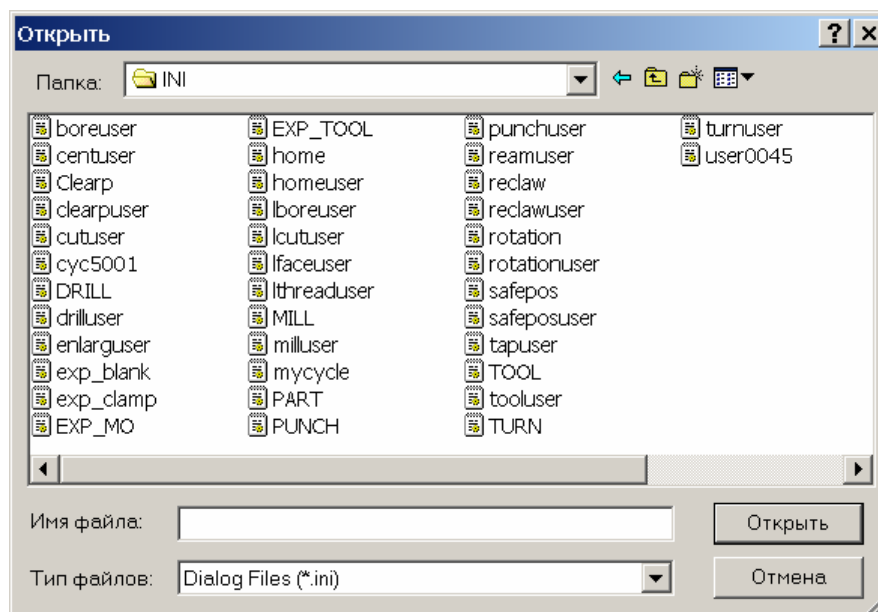
Элементы Меню:

Список	В окне отображается список элемент меню, состоящий из номера элемент меню и его имени. Чтобы установится на требуемый элемент меню, необходимо отметить его в списке.
Добавить	Добавить новый элемент меню в список. Элемент добавляется в конец списка.
Удалить	Удалить отмеченный элемент меню из списка.
Вверх	Переместить отмеченный элемент меню вверх списка.
Вниз	Переместить отмеченный элемент меню вниз списка.
Просмотр	используется для предварительного просмотра полученного меню.

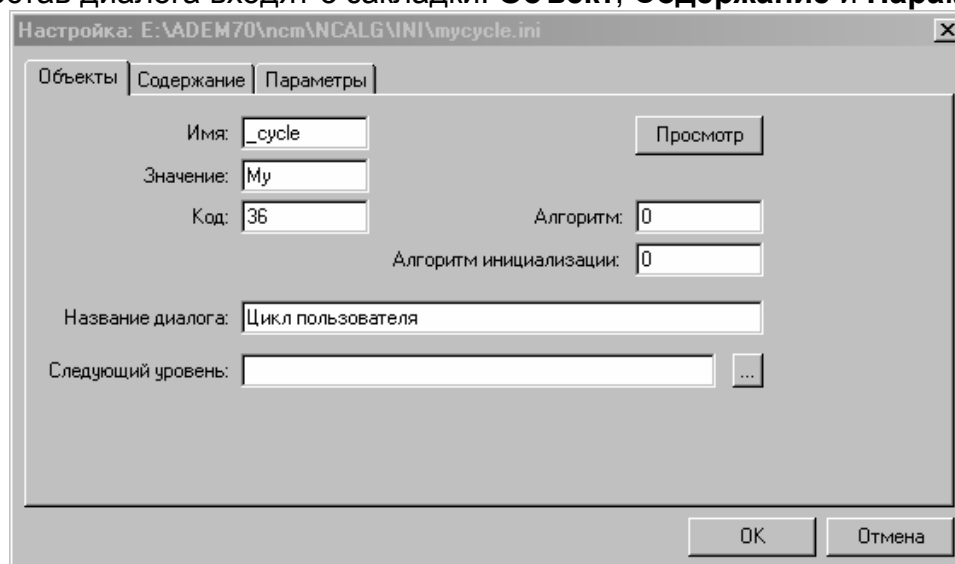
Элемент #1:

Имя	Имя элемента меню.
Следующий	Имя настроечного файла (*.ini) или файла меню (*.mnu). Устанавливает последовательность действий, которые будут произведены пользователем при выборе элемент меню. Если установлено имя настроечного файла, откроется диалог "Параметры Объекта", который формируется по шаблону данного настроечного файла. Если установлено имя файла меню, откроется следующее меню выбора, созданное на основе данного файла меню.

4. Добавьте новый цикл с именем «Расточка шпоночного паза», в качестве следующего укажите файл **mycycle_10.ini**.
5. Нажмите **ОК** и сохраните настроечный файл.
6. Повторите шаги 1 и 2, только теперь выберите тип файлов - *.ini, и откройте файл **mycycle.ini**.



7. В состав диалога входят 3 закладки: **Объект**, **Содержание** и **Параметры**.



Закладка «Объект»:

- | | |
|-----------------|---|
| Имя | Имя переменной, которая соответствует текущему объекту. Используется в алгоритмах. |
| Значение | Значение, которым инициализируется переменная, описанная параметром Имя (для пользовательских циклов – это номер цикла). |
| Код | Код объекта. Используется в алгоритмах для сортировки объектов по уровням (для пользовательских циклов всегда 36). |



Примечание

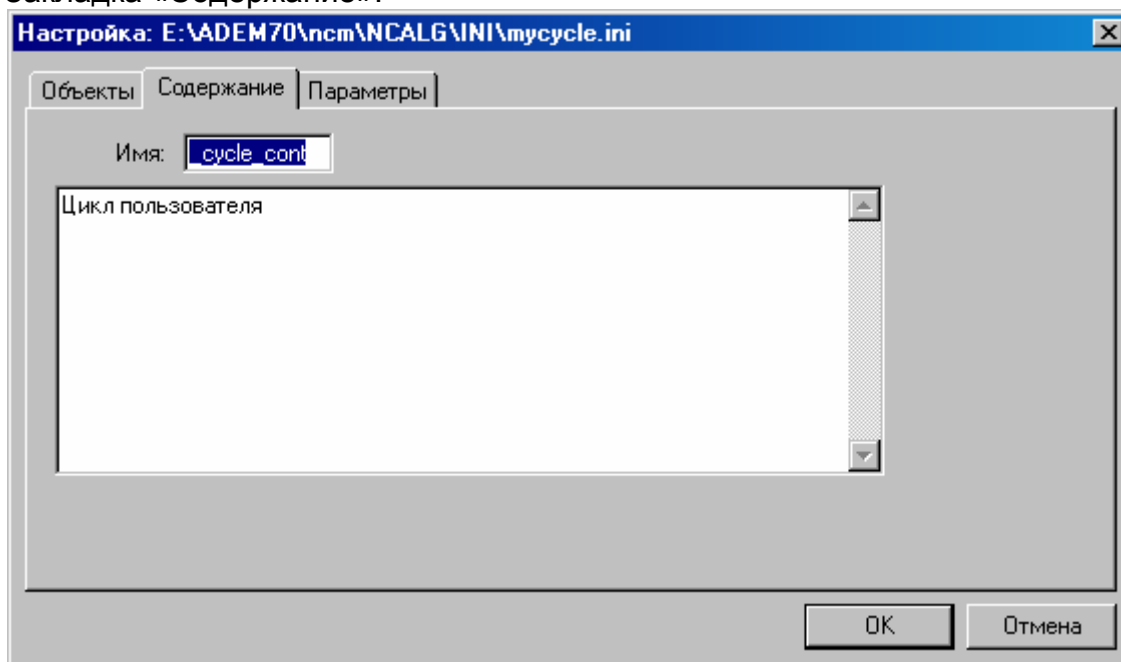
Объекты с кодом 99 имеют другой вид. В отличие от обычных объектов с кнопками «ОК» и «Отмена», у него только кнопка «ЗАКРЫТЬ»

- | | |
|-----------------|--|
| Алгоритм | Номер алгоритма. Если установлено не нулевое значение, |
|-----------------|--|

активизируется кнопка “Алгоритм”, при нажатии на которую выполняется требуемый алгоритм. Содержится алгоритм в файле с именем: 0000<номер алгоритма>.alg.

Алгоритм инициализации	Номер алгоритма инициализации. Если установлено не нулевое значение, при создании объекта выполняется требуемый алгоритм, который инициализирует параметры диалога. Содержится алгоритм в файле с именем: 0000<номер алгоритма>.alg.
Название диалога	Заголовок диалога.
Следующий уровень	Имя настроечного файла (*.ini) или файла меню (*.mnu). Устанавливает последовательность действий, которые будут произведены пользователем при создании объектов на следующем уровне. Если установлено имя настроечного файла, то при выполнении команды “Создать Объект” на следующем уровне откроется диалог “Параметры Объекта”, который формируется по шаблону данного настроечного файла. Если установлено имя файла меню, то при выполнении команды “Создать Объект” на следующем уровне откроется меню выбора, созданное на основе данного файла меню.
Просмотр	Используется для предварительного просмотра полученного диалога “Параметры Объекта”.

Закладка «Содержание»:



Имя	Имя переменной содержания. Используется в алгоритмах.
Текст содержания	Содержание используется для описания объекта. Текст является параметрическим. Связь с параметрами объекта устанавливается с помощью специального символа @, после которого устанавливается либо порядковый номер параметра, либо имя параметра, заключенное в [] скобки.

Закладка «Параметры»:

Список В окне отображается список параметров объекта, состоящий из номера параметра и имени его переменной. Чтобы установится на требуемый параметр, необходимо отметить его в списке.

Добавить Добавить новый параметр в список. Параметр добавляется в конец списка.

Удалить Удалить отмеченный параметр из списка.

Вверх Переместить отмеченный параметр вверх списка.

Вниз Переместить отмеченный параметр вниз списка.

Поле «Параметр @2»:

Имя Имя переменной текущего параметра. Используется в алгоритмах.

Значение Значение, которым инициализируется переменная параметра.

Заголовок Заголовок параметра.

Тип Тип параметра. Может принимать одно из следующих значений: *число*, *символ*, *дата*, *меню*, *меню2*, *вкладка*.

- *Число*. В качестве значения параметра могут использоваться математические выражения. Результат выполнения будет занесен в переменную параметра.
- *Дата*. Отображается стандартный управляющий элемент диалога для ввода даты.
- *Меню* и *меню2*. Комбинированный список, элементами которого являются строки текстового файла. Если определен тип *меню*, то после выбора строки в переменную параметра занесется число,

соответствующее порядковому номеру строки в файле. Если определен тип *меню2*, то – сама строка.

- Вкладка: Добавляет вкладку в диалог. В поле «Заголовок» необходимо ввести название вкладки. На вкладку помещаются все параметры находящиеся между двумя параметрами «Вкладка» или все параметры от текущего параметра «Вкладка» до конца. Если объект не имеет вкладок, то все элементы помещаются на вкладку – «Параметры».

Файл Имя текстового файла, связанного с данным окном, тип которого установлен как *меню* или *меню2*.

Алгоритм Номер алгоритма. Если установлено не нулевое значение, справа от значения появится кнопка “А”, при нажатии на которую выполняется требуемый алгоритм. Содержится алгоритм в файле с именем: *0000<номер алгоритма>.alg*.

Окно Тип окна вывода параметра. Может принимать одно из 3-х значений: *обычное, большое, скрытое, обычное только чтение, большое только чтение*. Тип окна *большое* могут принимать только нечетные параметры: 1-й, 3-й... Если установлен тип *скрытое*, параметр не будет отображаться в диалоге и не может корректироваться пользователем. Параметры, имеющие тип окна *обычное только чтение* и *большое только чтение* могут изменять свои значения только из алгоритмов, в режиме редактирования их значения изменить нельзя. Данный вид параметра диалога распространяются только на тип данных *число* и *строка*. На другие типы данных параметр окна *только чтение* игнорируется.

8. На закладке «Объекты» в поле «Название диалога» напишите: **Расточка шпоночного паза**, а в поле «Значение» - 10.

Настройка: E:\ADEM70\ncm\NCALG\INI\mycycle.ini

Объекты | Содержание | Параметры

Имя: Просмотр

Значение:

Код: Алгоритм:

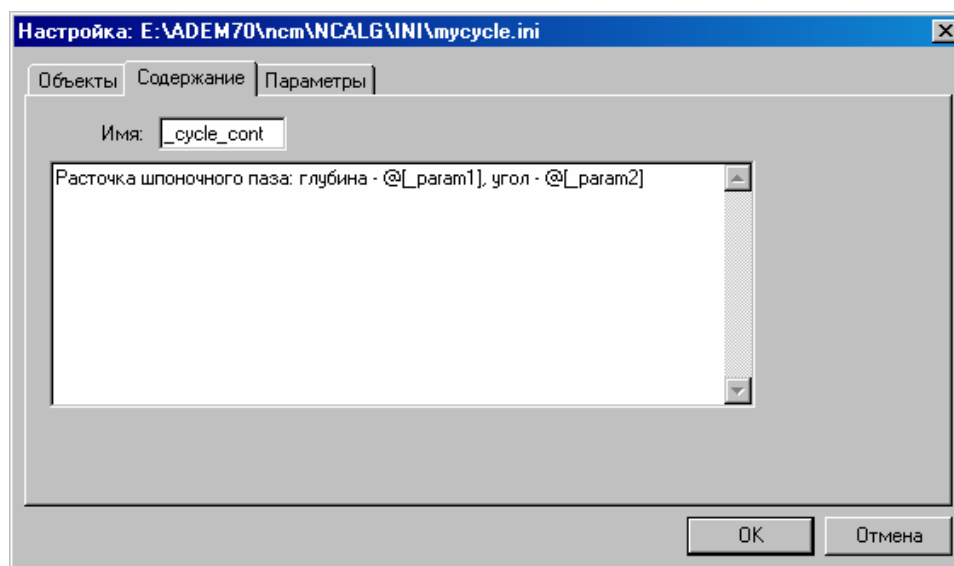
Алгоритм инициализации:

Название диалога:

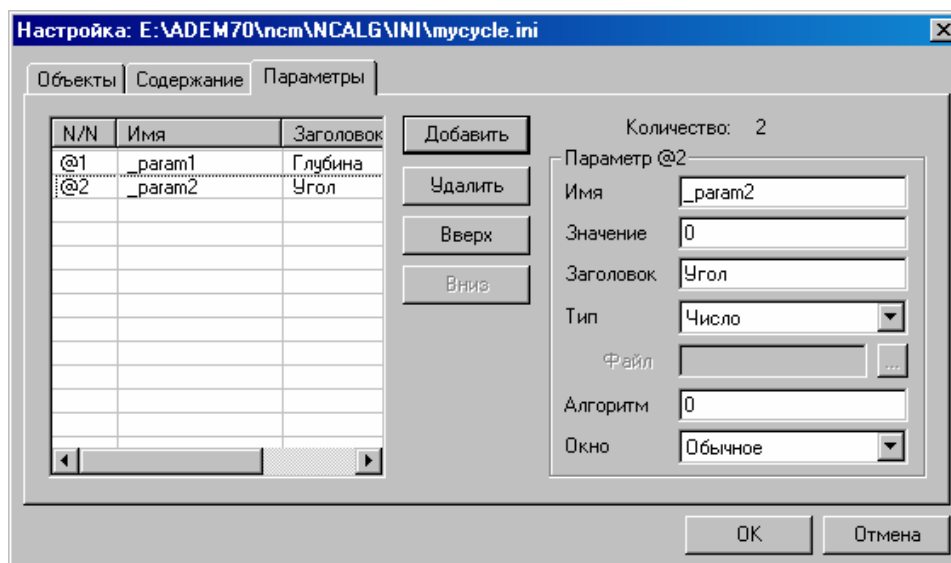
Следующий уровень: ...


OK Отмена

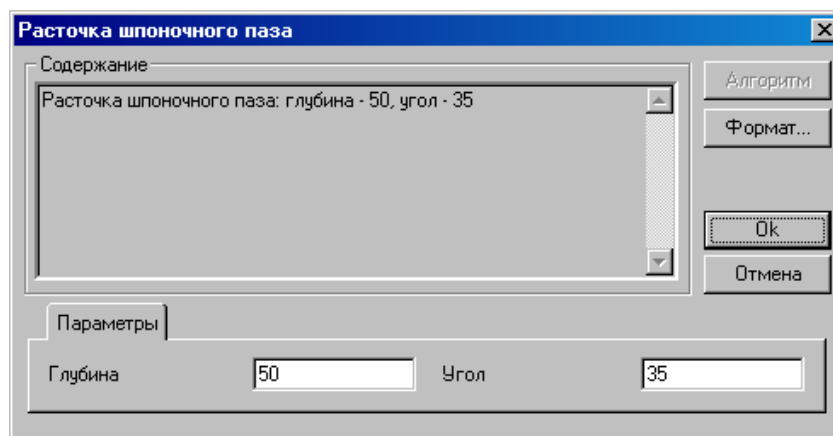
9. На закладке «Содержание» в поле «Текст содержания» напишите:
Расточка шпоночного паза: глубина - @[param1], угол - @[param2]



10. На закладке «Параметры» удалите все параметры и добавьте новые:
- **Глубина**
 - **Угол**



11. Нажмите **OK** и сохраните настроечный файл под именем **mycycle_10.ini**.
12. В модуле **ADEM CAM** нажмите кнопку  и выберите из контекстного меню пункт «Расточка шпоночного паза».



13. Введите параметры и нажмите **ОК**, после чего укажите окружность или дугу, центр которой будет являться точкой выполнения цикла. В маршруте обработки появится объект «**Расточка шпоночного паза**».

14. Для того, чтобы параметры цикла пользователя отразились в кадрах УП, в постпроцессоре необходимо описать порядок действий в алгоритме № 36.


```
36;
IF N/CYCLE=10 DO;
  _TXT->'ГЛУБИНА: @[PAR/C1]';
  BLOCK;
  _TXT->'УГОЛ: @[ PAR/C2]';
  BLOCK;
ENDDO;
END;
```

После формирования в кадрах УП Вы увидите следующее:


```
...
N2 ГЛУБИНА: 50
N3 УГОЛ: 35
...
```

6.5. Действия пользователя


Сформировать новый файл алгоритма

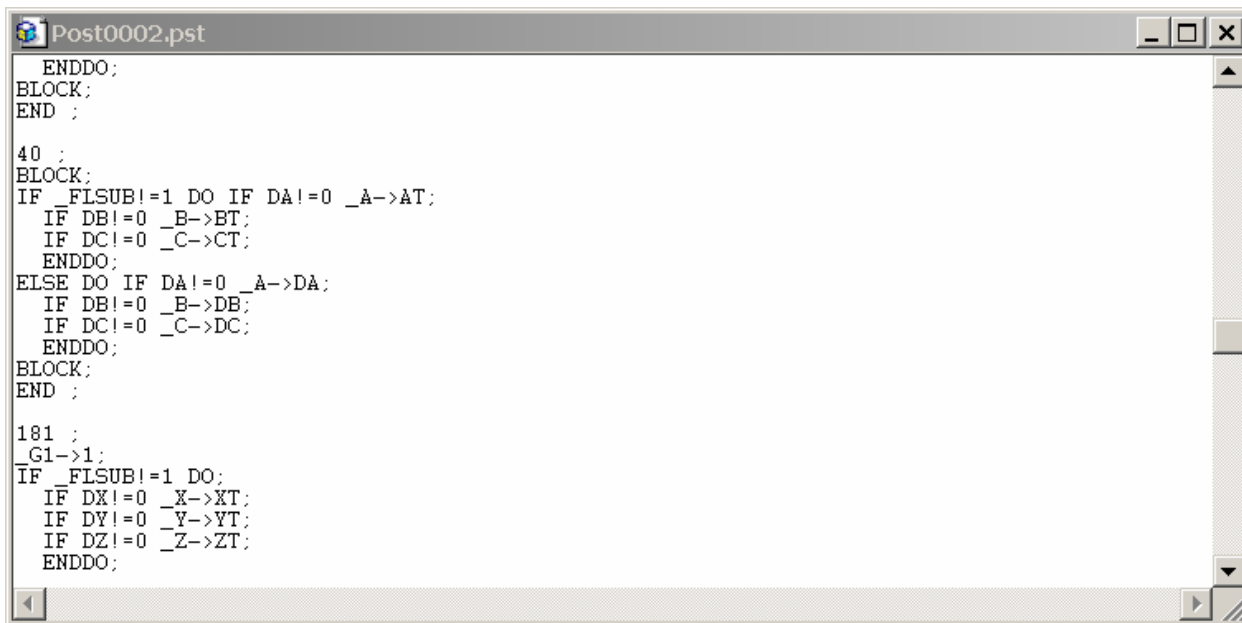
Нажмите кнопку **Создать** . Откроется окно с пустым текстовым файлом. При создании нового постпроцессора окно алгоритма создается автоматически.

Открыть алгоритм

Нажмите кнопку **Открыть**  и выберите файл алгоритма. Система откроет выбранный файл. При открытии постпроцессора окно алгоритма открывается автоматически.

Сохранить изменения в алгоритме

Нажмите кнопку **Сохранить**  Система сохранит изменения в текущем алгоритме. При сохранении постпроцессора и выполнения трансляции алгоритм сохраняется.



```

ENDDO;
BLOCK;
END ;

40 ;
BLOCK;
IF _FLSUB!=1 DO IF DA!=0 _A->AT;
  IF DB!=0 _B->BT;
  IF DC!=0 _C->CT;
ENDDO;
ELSE DO IF DA!=0 _A->DA;
  IF DB!=0 _B->DB;
  IF DC!=0 _C->DC;
ENDDO;
BLOCK;
END ;

181 ;
G1->1;
IF _FLSUB!=1 DO;
  IF DX!=0 _X->XT;
  IF DY!=0 _Y->YT;
  IF DZ!=0 _Z->ZT;
ENDDO;

```

Рис. 14. Текст алгоритма.

Алгоритм формируется и редактируется в текстовом виде. Перед трансляцией алгоритма в файл система проверяет его на наличие ошибок.

Примечание: • Строка не включается в алгоритм, если в ней допущены ошибки. Например, в конце нет символа ‘;’.

6.6. Пример формирования файла алгоритмов

Необходимо сформировать файл алгоритмов постпроцессора с номером 1 для токарного станка, т. е. занести в него алгоритмы следующих команд:

*Линейная интерполяция,
Круговая интерполяция,
Включить подачу,
Включить ускоренное перемещение,
Включить шпиндель,
Включить охлаждение,
Сменить инструмент,
Выключить шпиндель,
Выключить охлаждение,
Конец управляющей программы.*

Вызываем окно диспетчера постпроцессоров и устанавливаем его номер – 1. Можно начать с составления алгоритма на любую из команд CLDATA. Мы начнем с инициализации номеров окон макета кадра в алгоритме команды *Программа*, ее код равен 1. (Полный список команд с их кодами смотрите в [Приложении А](#).)

Предположим, макет кадра постпроцессора 1 имеет следующий вид:

N[G[X[Z[I[K[F[T[S[M[L[*]

Пользовательской переменной `_G` присвоим номер окна `G[]` в макете кадра. Для этого введем в алгоритм строку

```
_G=2;
```

Таким образом, инициализируем все номера окон макета кадра. Алгоритм после окончания набора примет следующий вид:

```
1;  
_G=2;  
_X=3;  
_Z=4;  
_I=5;  
_K=6;  
_F=7;  
_T=8;  
_SK=9;  
_M=10;  
_L=11;  
END;
```

Так как все номера окон макета инициализированы, закончим на этом составление алгоритма команды с кодом 1.

Вводим код команды *Линейная интерполяция*, ее код равен 181.

Предположим, что управляющая программа формируется в приращениях, приращение по оси `X` записывается в кадре с адресом `X`, приращение по оси `Z` записывается в кадре с адресом `Z`. Нулевые приращения по оси не формируются.

Сформируем алгоритм обработки команды. Чтобы сформировать ненулевое перемещение по оси `X`, занесем в окно `X[]` его величину. Для этого введем строку алгоритма:

```
IF DX!=0 _X->DX;
```

Аналогично сформируем ненулевое перемещение по оси `Z`:

```
IF DZ!=0 _Z->DZ;
```

Оформим информацию в виде кадра. Для этого введем команду алгоритма:

```
КАДР;
```

На экране:

```
181;  
IF DX!=0 _X->DX;  
IF DZ!=0 _Z->DZ;  
КАДР;  
END;
```

Вводим код команды *Круговая интерполяция*, ее код равен 183.

Предположим, что круговая интерполяция формируется в приращениях. Приращение по оси `X` записывается в кадре по адресу `X`, приращение по оси `Z` записывается в кадре по адресу `Z`, ненулевая абсолютная разница координат по оси `X` начальной точки дуги и центра дуги записываются по адресу `I`, ненулевая абсолютная разница координат по оси `Z` начальной точки дуги и центра дуги записываются по адресу `K`. Движение по часовой стрелке задается функцией `G2`, против часовой стрелки – `G3`.

Сформируем алгоритм:

```

183;
_X->DX;
_Z->DZ;
IF ХЦОКР!=ХС _I->ABS(ХЦОКР-ХС);
IF ZЦОКР!=ZC _K->ABS(ZЦОКР-ZC);
IF НАПРОКР=ЧС _G->2;
ELSE _G->3;
КАДР;
END;

```

Следующая команда: *Включить подачу*, код 23.

Предположим, что величина рабочей подачи формируется по адресу F. Сформируем алгоритм обработки команды:

```

23;
_F->S;
END;

```

Обратите внимание, что в алгоритме нет команды **КАДР**. Это значит, что информация в кадр будет выведена в последующих командах.

Следующая команда: *Включить ускоренное перемещение*, код 25.

Предположим, что величина ускоренного перемещения равна 70000 и формируется она по адресу F. Сформируем алгоритм обработки команды:

```

25;
_F->70000;
END;

```

Следующая команда: *Включить шпиндель*, код 24.

Предположим, что величина оборотов шпинделя задается по адресу S. Направление вращения по часовой стрелке определяет функция M3, против часовой стрелки – M4. Сформируем алгоритм обработки этой команды:

```

24;
_S->N;
IF НВШП=ЧС -M->3;
ELSE _M->4;
END;

```

Следующая команда: *Включить охлаждение*, код 26.

Предположим, что охлаждение включается функцией M8. Сформируем алгоритм обработки этой команды:

```

26;
_M->8;
END;

```

Следующая команда: *Сменить инструмент*, код 35.

Предположим, что номер позиции инструмента устанавливается по адресу T, а команда на загрузку инструмента задается функцией M6. Все должно быть оформлено отдельным кадром. Сформируем алгоритм обработки этой команды:

```

35;
КАДР;
_T->ТИНСТР;
_M->6;

```

```
КАДР;  
END;
```

Следующая команда: *Выключить шпиндель*, код 701.

Предположим, что выключение шпинделя задается функцией M5. Все должно быть оформлено отдельным кадром. Сформируем алгоритм отработки этой команды:

```
701;  
КАДР;  
_M->5;  
КАДР;  
END;
```

Следующая команда: *Выключить охлаждение*, код 700.

Предположим, что выключение охлаждения задается функцией M9. Все должно быть оформлено отдельным кадром. Сформируем алгоритм отработки этой команды:

```
700;  
КАДР;  
_M->9;  
КАДР;  
END;
```

Последняя команда: *Конец управляющей программы*, код 4.

Предположим, что конец управляющей программы задается функцией M2. Все должно быть оформлено отдельным кадром. Сформируем алгоритм отработки этой команды:

```
4;  
КАДР;  
_M->2;  
КАДР;  
END;
```

Итак, мы создали все необходимые алгоритмы

```
1;  
_G=2;  
_X=3;  
_Z=4;  
_I=5;  
_K=6;  
_F=7;  
_T=8;  
_S=9;  
_M=10;  
_L=11;  
END;  
  
181;  
IF DX!=0 _X->DX;  
IF DZ!=0 _Z->DZ;  
КАДР;  
END;
```

```
183;  
_X->DX;  
_Z->DZ;  
IF XЦОКР!=XC _I->ABS(XЦОКР-XC);  
IF ZЦОКР!=ZC _K->ABS(ZЦОКР-ZC);  
IF НАПРОКР=ЧC _G->2;  
ELSE _G->3;  
КАДР;  
END;  
23;  
_F->S;  
END;
```

```
25;  
_F->70000;  
END;
```

```
24;  
_S->N;  
IF НВШП=ЧC _M->3;  
ELSE _M->4;  
END;
```

```
26;  
_M->8;  
END;
```

```
35;  
КАДР;  
_T->ТИНСТР;  
_M->6;  
КАДР;  
END;  
701;  
КАДР;  
_M->5;  
КАДР;  
END;
```

```
700;  
КАДР;  
_M->9;  
КАДР;  
END;
```

```
4;  
КАДР;  
_M->2;  
КАДР;  
END;  
END;
```


После завершения процесса формирования файла алгоритмов сохраним созданный файл постпроцессора под номером: 1.

Глава 7. Системные переменные.

Напомним, что *системные переменные* автоматически принимают значения параметров команд CLDATA, значение *пользовательских переменных* изменяется только Вами.

- Внимание!**
- Системные переменные имеют фиксированные имена.
 - В ранних версиях ADEM GPP использовались фиксированные имена для пользовательских переменных. Сейчас достаточно поставить перед именем переменной знак подчеркивания «_» и она будет интерпретироваться адаптером как пользовательская.

Данная глава состоит из 25 разделов, которые содержат описание системных переменных. В 25 разделе указаны имена *пользовательских переменных*, которые использовались в ранних версиях ADEM GPP.

7.1. Координаты инструмента

Текущая точка (ТТ) – это точка, в которой находится инструмент.

Предыдущая точка (ПТ) – это точка, из которой произошло перемещение в ТТ.

ХТ XCURPOS	Координата X ТТ.
УТ YCURPOS	Координата Y ТТ.
ЗТ ZCURPOS	Координата Z ТТ.
ХС XBEG	Координата X ПТ.
УС YBEG	Координата Y ПТ.
ЗС ZBEG	Координата Z ПТ.
DX	Перемещение по оси X, равное ХТ–ХС.
DY	Перемещение по оси Y, равное УТ–УС.
DZ	Перемещение по оси Z, равное ЗТ–ЗС.
АТ	Угловая координата по оси А ТТ.

ACURPOS	
BT BCURPOS	Угловая координата по оси В ТТ.
CT CCURPOS	Угловая координата по оси С ТТ.
AC ABEG	Угловая координата по оси А ПТ.
BC BBEG	Угловая координата по оси В ПТ.
CC CBEG	Угловая координата по оси С ПТ.
DA	Угловое перемещение по оси А, равное АТ–АС.
DB	Угловое перемещение по оси В, равное ВТ–ВС.
DC	Угловое перемещение по оси С, равное СТ–СС.
ДЛПЕР LENG/MOV	Длина перемещения.

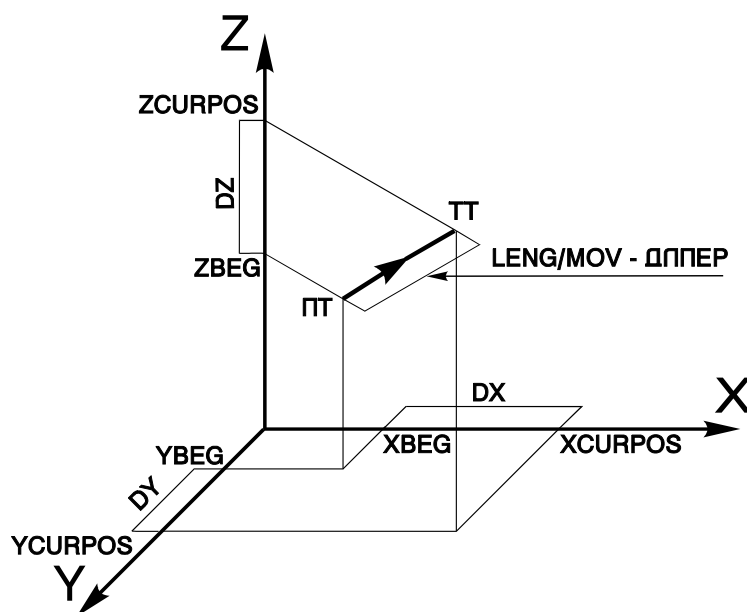


Рис. 11. Параметры линейного перемещения.

Этим системным переменным присваиваются значения параметров при обработке следующих команд CLDATA:

- *Линейная интерполяция* (код 181);
- *Круговая интерполяция* (код 183);
- *Совмещенное перемещение* (код 41);
- *Поворот* (код 40).

7.2. Круговая интерполяция

Точка центра окружности (ТЦО) – это центр дуги, по которой произошло перемещение из ПТ в ТТ.

XЦОКР X/CENTER	Координата X ТЦО.
YЦОКР Y/CENTER	Координата Y ТЦО.
ZЦОКР Z/CENTER	Координата Z ТЦО.
ROKР R/CIRC	Радиус дуги, по которой произошло перемещение из ПТ в ТТ.
НАПРОКР DIR/CIRC	Направление движения по дуге. Принимает значение ЧС или ПЧС.
ЧС CW	Движение по часовой стрелке.
ПЧС CWW	Движение против часовой стрелки.
ВОКР TYP/CIRC	Вид дуги, по которой происходит перемещение, принимает значение ВЫП или ВОГН.
ВЫП CONVEX	Выпуклая дуга.
ВОГН CONCAVE	Вогнутая дуга.
ЦУГОЛ ANG/CIRC	Центральный угол дуги, по которой происходит круговая интерполяция.
КПЛОК ПЛОКР C/PLACE	Плоскость движения по дуге окружности, принимает значения XY, YZ или ZX.
XY YX	Плоскость XY.
YZ ZY	Плоскость YZ.
ZX XZ	Плоскость ZX.
CIRC/ON CIRC/OFF	Включение и выключение линейных интерполяций.

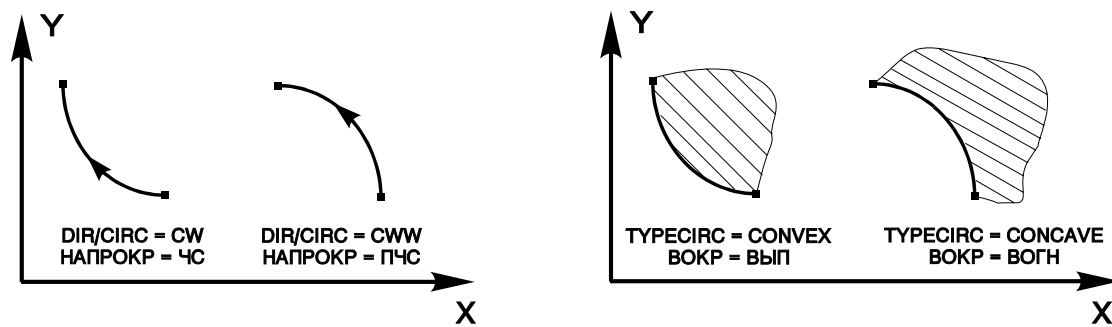


Рис. 12. Параметры кругового перемещения

7.3. Последующие перемещения инструмента

ТСЛ (следующая точка) – точка обработки детали, в которую произойдет следующее перемещение инструмента.

Смотрите рисунок 5.

ВХСЛ V/XNEXT	Вектор перемещения по оси в следующую точку обработки детали. Если следующее перемещение по траектории обработки – линейная интерполяция, ВХСЛ принимает значение координаты X ТСЛ. В случае круговой интерполяции, ВХСЛ принимает значение косинуса касательной к окружности, по которой будет происходить перемещение, в точке начала дуги.
ВУСЛ V/YNEXT	Вектор перемещения по оси в следующую точку обработки детали. Если следующее перемещение – линейная интерполяция, ВХСЛ принимает значение координаты Y ТСЛ. В случае круговой интерполяции, ВХСЛ принимает значение синуса касательной к окружности, по которой будет происходить перемещение, в точке начала дуги.
ВЗСЛ V/ZNEXT	Вектор перемещения по оси в следующую точку обработки детали. Если следующее перемещение – линейная интерполяция, ВХСЛ принимает значение координаты Z ТСЛ.
УГНАП ANG/M	Угол перелома траектории (в радианах).
К/ПЕРЕЛ C/BREAK	Код перелома траектории. Принимает значение ВНЕШ при внешнем обходе угла, ВНУТ при обработке внутреннего угла, в случае сопряжения К/ПЕРЕЛ=-1.
XNEXT	Координата X следующего перемещения.
YNEXT	Координата Y следующего перемещения.
ZNEXT	Координата Z следующего перемещения.
XCNEXT	Координата X центра дуги следующего перемещения, если следующее перемещение – круговая интерполяция.
YCNEXT	Координата Y центра дуги следующего перемещения, если следующее перемещение – круговая интерполяция.
ZCNEXT	Координата Z центра дуги следующего перемещения, если следующее перемещение – круговая интерполяция.
RNEXT	Радиус дуги следующего перемещения, если следующее перемещение – круговая интерполяция.

CNEXTM	Код следующего перемещения. Принимает следующие значения: 181 – линейное перемещение, 183 – круговая интерполяция, 35 – после текущего перемещения встречена смена инструмента, 4 – после текущего перемещения обнаружен конец программы.
CCIRCM	Направление движения по дуге следующего перемещения, если следующее перемещение – круговая интерполяция. Принимает значения ЧС или ПЧС.

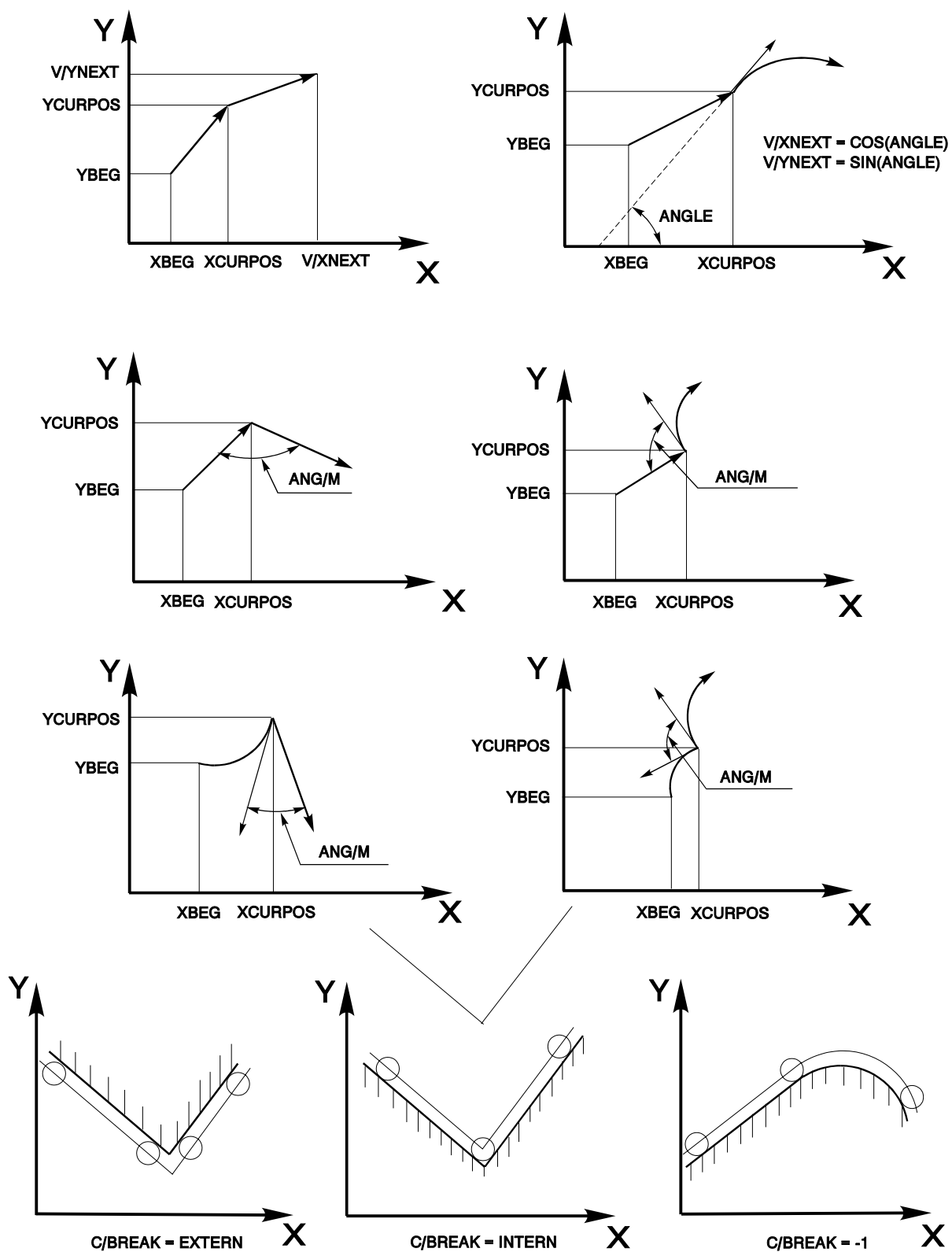


Рис. 14. Переменные, характеризующие траекторию обработки

7.4. Совмещенные перемещения.

XCUR2	Координата X точки совмещенного перемещения.
YCUR2	Координата Y точки совмещенного перемещения.
ZCUR2	Координата Z точки совмещенного перемещения.
XOLD2	Координата X начальной точки совмещенного перемещения.
YOLD2	Координата Y начальной точки совмещенного перемещения.
ZOLD2	Координата Z начальной точки совмещенного перемещения.
X/CENT2	Координата X центра дуги совмещенного перемещения, если совмещенное перемещение проходит по дуге.
Y/CENT2	Координата Y центра дуги совмещенного перемещения, если совмещенное перемещение проходит по дуге.
Z/CENT2	Координата Z центра дуги совмещенного перемещения, если совмещенное перемещение проходит по дуге.
DIR/CIR2	Направление движения по дуге совмещенного перемещения, если совмещенное перемещение проходит по дуге. Принимает значения ЧС или ПЧС.

7.5. Геометрия и номер позиции инструментов

ТИНСТР N/TOOL	Номер позиции загружаемого инструмента.
CURTOOL	Номер позиции текущего инструмента.
ИНСТР1 FIRSTOOL	Номер позиции первого инструмента в программе.
СЛИНСТР NEXTOOL	Номер позиции инструмента, который будет загружен следующим по ходу программы. Если загружен последний инструмент в программе, значение СЛИНСТР формируется в зависимости от постпроцессора: при необходимости загрузки первого инструмента в конце программы СЛИНСТР = ИНСТР1, в противном случае СЛИНСТР = 0.
КОЛИНСТ NOF/TOOL	Количество инструментов в таблице.
ПНИНСТ IND/TOOL	Порядковый номер загружаемого инструмента в таблице.
РИНСТР RAD/TOOL	Радиус загружаемого инструмента.

РСЛИНСТР NEXT/RTL	Радиус инструмента, который будет загружен следующим в программе.
ВЫЛЕТХ XOVERH	Вылет инструмента по оси X.
ВЫЛЕТУ YOVERH	Вылет инструмента по оси Y.
ВЫЛЕТZ ZOVERH	Вылет инструмента по оси Z.
НПИТ NPT/TBL	Номер позиции инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
NKXT XCT/TBL	Номер корректора X инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
NKYT NKYT YCT/TBL	Номер корректора Y инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
NKZT ZCT/TBL	Номер корректора Z инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
NKRT RCT/TBL	Номер радиусного корректора инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
ТВЫЛЕТХ	Величина вылета по оси X инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
ТВЫЛЕТУ ТВЫЛЕТУ	Величина вылета по оси Y инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
ТВЫЛЕТZ	Величина вылета по оси Z инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
ТДИАМ TABLDIAM	Диаметр инструмента из таблицы. Устанавливается после отработки команды алгоритма СТИНСТ(RTOOLPAR).
ANG/TOOL	Угол инструмента в радианах.
L/TOOL	Длина режущей части инструмента.
TL/TOOL	Общая длина инструмента.
COD/TOOL	Код загружаемого/загруженного инструмента. Принимает следующие значения:
	Инструмент
	Фреза
	Сверло
	Код
	1
	2

	Центровка	3
	Зенкер	4
	Развертка	5
	Метчик	6
	Резец	7
	Пуансон	8
	Проволока	9
	Лазер	10
	Резак	11
N/TUR	Номер турреты, для двухтурретной обработки	
N/CP	Номер контрольной точки при многотурретной обработке.	

7.6. Включение/выключение корректоров

ВКЛКОРХ XCOMPON	Номер включаемого линейного корректора по оси X. После отработки команды КХ/ВКЛ значение ВКЛКОРХ сбрасывается.
ВКЛКОРУ YCOMPON	Номер включаемого линейного корректора по оси Y. После отработки команды КУ/ВКЛ значение ВКЛКОРУ сбрасывается.
ВКЛКОРZ ZCOMPON	Номер включаемого линейного корректора по оси Z. После отработки команды КZ/ВКЛ значение ВКЛКОРZ сбрасывается.
ВКЛКОРР RCOMPON	Номер включаемого радиусного корректора. После отработки команды КR/ВКЛ значение ВКЛКОРР сбрасывается.
ВЫКЛКОРХ XCOMPOFF	Номер выключаемого линейного корректора по оси X. После отработки команды КХ/ВЫКЛ значение ВЫКЛКОРХ сбрасывается.
ВЫКЛКОРУ YCOMPOFF	Номер выключаемого линейного корректора по оси Y. После отработки команды КУ/ВЫКЛ значение ВЫКЛКОРУ сбрасывается.
ВЫКЛКОРZ ZCOMPOFF	Номер выключаемого линейного корректора по оси Z. После отработки команды КZ/ВЫКЛ значение ВЫКЛКОРZ сбрасывается.
ВЫКЛКОРР RCOMPOFF	Номер выключаемого радиусного корректора. После отработки команды КR/ВЫКЛ значение ВЫКЛКОРР сбрасывается.

7.7. Выстой

ВЫСТОБ	Величина выстой, выраженная в количестве оборотов
--------	---

SPIN/DW	шпинделя.
ВЫСТВР TIME/DW	Величина выстоя, выраженная в секундах.
КОДВЫСТ C/DWELL	Показывает, каким образом была задана величина выстоя при формировании перехода технологического объекта. Принимает значения 0, если выстой был задан в секундах или 1, если выстой был задан в оборотах шпинделя. Вне зависимости от способа задания выстоя его величина рассчитывается и оборотах и в секундах и записывается в системные переменные ВЫСТОБ(SPIN/DW) и ВЫСТВР(TIME/DW) соответственно.

Эти параметры формируются по команде ВЫСТОЙ (код 27).

7.8. Положение металла

ПЛМ METLOCAT	Положение металла. ПЛМ принимает значения СЛВ или СПР.
СЛВ LEFT	Металл слева [1].
СПР RIGHT	Металл справа [0].

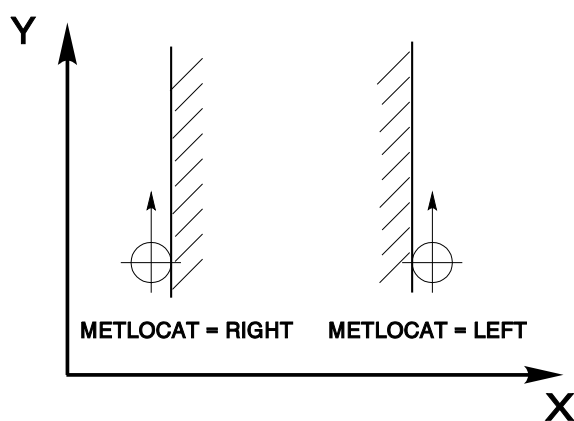


Рис. 15. Параметры положения металла при обработке

7.9. Управление шпинделем

N NOB SPIN	Величина оборотов шпинделя в об/мин.
V VPEЗ	Скорость резания в м/мин.

VC	
НДИАП N/RANGE	Номер заданного диапазона шпинделя.
НВШП DIR/SPIN	Направление вращения шпинделя, принимает значение ЧС или ПЧС.
КОДШПИНД COD/SPIN	Код в ряду шпинделя.
СЛНОВБ NEXTSPIN	Величина оборотов шпинделя, которая будет включена далее по ходу программы в об/мин.
СЛVPEЗ NEXTVC	Величина СЛНОВБ, пересчитанная в м/мин.
СЛНДИАП NEXT/RNG	Номер диапазона шпинделя, который будет включен далее по ходу программы.
СЛНВШП NX/DIRSP	Направление вращения шпинделя, которое будет включено далее по ходу программы, принимает значение ЧС или ПЧС.
СЛКОДШП NEXTC/SP	Следующий код в ряду шпинделя.
УГОРШП ANG/SPIN	Угол ориентации шпинделя при ориентированном останове.
СЛККШП	Показывает, каким образом будут заданы обороты шпинделя при смене текущей загрузки шпинделя. Принимает значения об/мин или м/мин. Вне зависимости от способа задания оборотов шпинделя его величина рассчитывается и в оборотах в минуту и в метрах в минуту и записывается в системные переменные СЛНОВБ(NEXTSPIN) и СЛVPEЗ(NEXTVC) соответственно.
ККШП	Показывает, каким образом были заданы обороты шпинделя при формировании перехода технологического объекта. Принимает значения об/мин или м/мин. Вне зависимости от способа задания оборотов шпинделя его величина рассчитывается и в оборотах в минуту и в метрах в минуту и записывается в системные переменные N(SPIN) и V(VC) соответственно.
MAX/SPIN	Максимально допустимая величина оборотов шпинделя (в токарной обработке).

7.10. Управление подачей

S ПОДМИН FEED	Величина подачи в мм/мин.
СОБ ПОДОБ	Величина подачи в мм/об.

FEEDS/T	
СЛПОДМИН NEXT/F/T	Величина подачи, которая будет включена позже в мм/мин.
СЛПОДОБ NEXT/FD	Величина слподмин, пересчитанная в об/мин.
КЗПОД CSETF	Показывает, каким образом была задана подача при формировании перехода технологического объекта. Принимает значения мм/мин или мм/об. Вне зависимости от способа задания подачи ее величина рассчитывается и в миллиметрах в минуту и в миллиметрах на оборот и записывается в системные переменные S(FEED) и SOB(FEEDS/T) соответственно.

7.11. Резьба

П/РЕЗЬБЫ PR/THRD	Профиль резьбы, принимает значение МЕТРИЧ, ТРАПЕЦ, УПОРНАЯ, ТРУБНАЯ или ПРЯМОУГ.
МЕТРИЧ METRIC	Метрическая [0].
ТРАПЕЦ TRAPEZ	Трапецеидальная [1].
УПОРНАЯ BUTTRESS	Упорная [2].
ТРУБНАЯ PIPE	Трубная [3].
ПРЯМОУГ SQUARE	Прямоугольная [4].
В/РЕЗЬБЫ KINDTHRD	Вид резьбы, принимает значение НАРУЖ или ВНУТ.
НАРУЖ EXTERN	Наружная [0].
ВНУТ INTERN	Внутренняя [1].
Т/РЕЗЬБЫ TYPETHRD	Тип резьбы, принимает значение ПРАВАЯ или ЛЕВАЯ.
ПРАВАЯ RIGHT	Правая [0].
ЛЕВАЯ LEFT	Левая [1].
СБЕГ RUN/OUT	Сбег резьбы, принимает значение ВКЛ или ВЫКЛ.
ВКЛ	Включен [0].

ON	
ВЫКЛ OFF	Выключен [1].
Ф/РЕЗЬБЫ FORMTHRD	Форма резьбы, принимает значение ЦИЛ или КОНИЧ.
ЦИЛ STRAIGHT	Цилиндрическая [0].
КОНИЧ TAPER	Коническая [1].
УГР ANG/THRD	Угол резьбы (в радианах).
ШАГР PITCH/T	Шаг резьбы.
ДЛИНР LENGTH	Длина резьбы.
NOF/THST КПРОХР	Количество проходов при нарезании резьбы.

7.12. Учетные параметры программы, детали и станка

НПРОГ N/PROG	Номер программы, формируемый из имени программы. Если имя программы не число, НПРОГ = 1.
ИПРОГ NAMEPROG	Имя программы (текстовая информация).
ИДЕТ NAMEPART	Имя детали (текстовая информация).
НДЕТ N/PART	Номер детали (текстовая информация).

7.13. Постоянные циклы

НЦИКЛ N/CYCLE	Номер цикла.
NEXTCYCL	Номер цикла, который будет отрабатываться после текущей команды CL DATA.
НЦИКЛ N/CYCLE	Номер цикла.
КПАРЦ NOF/PARC	Количество параметров цикла.
ПАРЦ1...ПАРЦ20	Параметры цикла.

PAR/C1...PAR/C20

КОЛТЦ Кол-во точек сверлильно-расточных циклов.
NOF/DMC

Параметры стандартных сверлильно–расточных циклов:

ПАРЦ1 Подача цикла.
PAR/C1
CYC/FEED

ПАРЦ2 Плоскость отвода инструмента.
PAR/C2
CYC/VRET

ПАРЦ3 Глубина отверстия.
PAR/C3
CYC/VDEP

ПАРЦ4 Пауза.
PAR/C4
CYC/DWEL

ПАРЦ5 Число заглублений минус 1.
PAR/C5
CYC/STEP

ПАРЦ8 Приращение при глубоком сверлении.
PAR/C8
CYC/PECK

7.14. Координаты безопасной позиции

ХБЕЗПОЗ Координата X безопасной позиции.
X/GOHOME

УБЕЗПОЗ Координата Y безопасной позиции.
Y/GOHOME

ZБЕЗПОЗ Координата Z безопасной позиции.
Z/GOHOME

7.15. Координаты точки прижима

ХПРИЖ Координата X точки прижима (ТПР).
XPRESS

УПРИЖ Координата Y ТПР.
YPRESS

СМЗАЖ Смещение зажима.
CLAMPMOV

7.16. Номер стола

НСТОЛ N/TABLE	Номер стола.
------------------	--------------

7.17. Номер трубопровода СОЖ

НСОЖ N/COOL	Номер трубопровода СОЖ.
----------------	-------------------------

7.18. Начало цикла

Если начало цикла задано координатами, следующие переменные принимают значения:

ХНЦ XHOME	Координата X точки начала цикла (ТНЦ).
YНЦ YHOME	Координата Y ТНЦ.
ZНЦ ZHOME	Координата Z ТНЦ.

Если НЦ задано регистром, присваивается значение переменной:

ННЦ N/HOME	Номер регистра с координатами НЦ.
---------------	-----------------------------------

Если НЦ задано корректорами, присваиваются значения переменным:

KXНЦ XCHOME	Номер корректора по оси X.
KYНЦ YCHOME	Номер корректора по оси Y.
KZНЦ ZCHOME	Номер корректора по оси Z.
КОЛНЦ NOF/HPOS	Количество определенных НЦ в таблице.
XТТЦ X/TPC	Координата X точки из таблицы. Устанавливается после отработки команды алгоритма СКТЦ(RPCYC).
YТТЦ Y/TPC	Координата Y точки из таблицы. Устанавливается после отработки команды алгоритма СКТЦ(RPCYC).
ZТТЦ Z/TPC	Координата Z точки из таблицы. Устанавливается после отработки команды алгоритма СКТЦ(RPCYC).
СПНЦ RN/HOME	Способ задания НЦ, записанного в таблицу. Устанавливается после отработки команды алгоритма

СТНЦ(RHOMЕ). СПНЦ=0, если НЦ было задано номером системы координат, СПНЦ=1, если НЦ было задано номерами корректоров, иначе НЦ было задано координатами X,Y и Z.

ХНЦТ Х/ТРН ННЦТ	Устанавливается после отработки команды алгоритма СТНЦ(RHOMЕ). Если НЦ было задано номером системы координат, переменная содержит номер системы координат, если НЦ было задано номерами корректоров, переменная содержит номер корректора X, иначе переменная содержит координату X точки НЦ.
УНЦТ УНЦТ У/ТРН	Устанавливается после отработки команды алгоритма СТНЦ(RHOMЕ). Если НЦ было задано номером системы координат, переменная содержит ноль, если НЦ было задано номерами корректоров, переменная содержит номер корректора Y, иначе переменная содержит координату Y точки НЦ.
ЗНЦТ Z/ТРН	Устанавливается после отработки команды алгоритма СТНЦ(RHOMЕ). Если НЦ было задано номером системы координат, переменная содержит ноль, если НЦ было задано номерами корректоров, переменная содержит номер корректора Z, иначе переменная содержит координату Z точки НЦ.
NOF/INC КТЦТ	Количество точек выполнения группы циклов. Устанавливается после отработки команды алгоритма СКТЦТ (RQPCYC).
ПНЦТ IND/CP	Порядковый номер текущей команды CLDATA ЦИКЛ (код 36). Учитываются только стандартные циклы (с номерами 81-89).
NKXНЦ	Номер корректора X НЦ
NKYНЦ	Номер корректора Y НЦ
NKYНЦ	
NKZНЦ	Номер корректора Z НЦ
NEXTCYCL	Номер следующего цикла.
QP/DMS КОЛТЦ	Общее количество точек циклов в таблице, выводятся только неповторяющиеся значения.
NUM/PC НТТЦ	Порядковый номер цикла в маршруте обработки.

7.19. Работа с подпрограммами.

НПОДПР	Номер вызываемой подпрограммы. Формируется на основе имени проекта, использованного как подпрограмма. Если имя не число, эта переменная принимает значение 1.
НПОДПР	
N/SUB	

КПОВТППГ	Количество повторений подпрограммы при вызове.
НФППГ	Номер формируемой подпрограммы. Формируется на основе имени проекта, использованного как подпрограмма.
NCRSUB	Если имя не число, эта переменная принимает значение 1.
КПАРППГ	Количество параметров вызываемой подпрограммы.
ПАРПП1... ...ПАРПП20	Величины параметров вызываемой подпрограммы.
XFSUB	Координата X первого перемещения вызываемой подпрограммы.
YFSUB	Координата Y первого перемещения вызываемой подпрограммы.
ZFSUB	Координата Z первого перемещения вызываемой подпрограммы.
ИБППГ	Имя вызываемой подпрограммы. Формируется на основе имени проекта, использованного как подпрограмма. Выводится в окно типа "ТЕКСТ".
ИФППГ	Имя формируемой подпрограммы. Формируется на основе имени проекта, использованного как подпрограмма. Выводится в окно типа "ТЕКСТ".

7.20. Системные переменные для работы с контурами и CLData.

COEFMM	Коэффициент, показывающий текущие единицы измерения. Если COEFMM=1, то текущие единицы измерения миллиметры, а если COEFMM=2,54 – дюймы.
XCLDMIN	Минимальная координата X, полученная в CLData.
YCLDMIN	Минимальная координата Y, полученная в CLData.
ZCLDMIN	Минимальная координата Z, полученная в CLData.
XCLDMAX	Максимальная координата X, полученная в CLData.
YCLDMAX	Максимальная координата Y, полученная в CLData.
ZCLDMAX	Максимальная координата Z, полученная в CLData.
RTOOLMAX	Максимальный радиус инструмента, используемого в маршруте обработки.
XCONTMIN	Минимальная координата X контура обработки.
YCONTMIN	Минимальная координата Y контура обработки.

XCONTMAX	Максимальная координата X контура обработки.
YCONTMAX	Максимальная координата X контура обработки.
PLANEMAX	Максимальная плоскость конструктивного элемента, заданная в маршруте обработки.
FR[]	Массив, содержащий элементы текущей фразы CLData.
FR[i]	i-тый элемент текущей фразы CLData.

7.21. Системные переменные для работы с пользовательскими функциями.

N/USFUNC	Номер функции пользователя.
NEXTUSC	Номер функции пользователя, которая будет отрабатываться после текущей команды CL DATA.
PUSFUN1- PUSFUN20	Параметры функции пользователя.
X/COORD	Координата X пользовательской системы координат.
Y/COORD	Координата Y пользовательской системы координат.
Z/COORD	Координата Z пользовательской системы координат.
A/COORD	Угол пользовательской системы координат.
NEXTUSC	Номер следующей пользовательской функции.

7.22. 5-ти координатные перемещения и работа с трансформами.

VSP/X	Направляющие косинусы для пяти-координатного перемещения инструмента.
VSP/Y	
VSP/Z	
VSNP/X	Направляющие косинусы для нормали к поверхности в точке касания.
VSNP/Y	
VSNP/Z	
ASP1, ASP2	Углы для пяти-координатного перемещения (в радианах).
SYSCOOR	Единичная (базовая) система координат.
CLDCOOR	Текущая система координат (из CLData).
RCOOR	Трансформ точки поворота.
COORUS1...	Промежуточные системы координат для вычислений.

... COORUS10

CALCOOR Трансформ для пересчета координат точек CLData в координаты точек управляющей программы.

7.23. Системные переменные для работы с элементами контуров.

QTY/EL	Количество элементов в контуре.
CODE/EL	Код элемента контура: 2 – отрезок; 3 – дуга.
XBEG/EL	Начальная координата X элемента контура.
YBEG/EL	Начальная координата Y элемента контура.
ZBEG/EL	Начальная координата Z элемента контура.
XEND/EL	Конечная координата X элемента контура.
YEND/EL	Конечная координата Y элемента контура.
ZEND/EL	Конечная координата Z элемента контура.
DIRC/EL	Направление движения по дуге, принимает значения ЧС и ПЧС.
RCIRC/EL	Радиус дуги.
XCEN/EL	Координата X центра дуги.
YCEN/EL	Координата Y центра дуги.
ZCEN/EL	Координата Z центра дуги.

7.24. Вспомогательные переменные

НОМКДР N/BLOCK	Номер кадра.
ТКОМ REM	Комментарий.
КСЛКОМ NEXT/COD	Код следующей команды CLDATA.
КОТВОД C/GOHOME	Код отвода. КОТВОД = ВКЛ, если генерируются команды отвода инструмента. КОТВОД = ВЫКЛ, – отработка команды CLDATA.
НПОДВОД BEGSTART	Флаг начала подвода. Имеет значение НПОДВОД=ВКЛ, если отрабатываются специальные команды подвода (см. команду ПОДВОД). В остальных случаях НПОДВОД=ВЫКЛ.

ПННЦ IND/HPOS TIME	Порядковый номер текущей команды CLDATA НЦ (код 401). Переменная, которая содержит время работы управляющей программы при выключенном условии автоматического расчета времени работы управляющей программы адаптером. Для отключения автоматического времени расчета необходимо дать команду РАСЧВР=ВЫКЛ; , а в переменной TIME накапливать время работы УП.
AUTOTIME РАСЧВР	Переменная, которая содержит признак автоматического расчета времени работы УП. Начальное значение переменной РАСЧВР=ВКЛ. Чтобы иметь возможность рассчитывать время работы УП в постпроцессоре, необходимо дать команду РАСЧВР=ВЫКЛ; а время работы сохранять в переменной TIME.
TBCT INS/TEXT	Текст вставки.
SIZEFILE	Считает количество символов, выведенных в кадры управляющей программы.
DAY	Текущий день недели
MONTH	Текущий месяц
YEAR	Текущий год
HOURL	Текущий час
MINUTE	Текущая минута
SECOND	Текущая секунда

7.25. Пользовательские переменные

P1...P10
 G1(G)...G20
 NK
 E
 X(X1), X2
 Y(Y1), Y2
 Z(Z1), Z2
 D(D1)...D3
 I, J, K

T, F, SK
 L(L1)...L3
 A, B, C
 H(H1)...H3
 M(M1)...M3
 R
 Q
 LF


Примечание: • В скобках даны дополнительные имена переменных.
 Например, G и G1 - одна и та же переменная. □

Напомним, что *системные переменные* автоматически принимают значения параметров команд CLDATA, значение *пользовательских переменных* изменяется только Вами.

Глава 8. Отладчик алгоритмов

Все этапы формирования адаптером управляющей программы можно пронаблюдать при запуске процессора адаптации в режиме отладчика.

Для этого необходимо:

- В модуле ADEM CAM установить имя станка (получить CLDATA);
- Войти в модуль ADEM GPP;
- Запустить отладчик (кнопка **Отладка** )

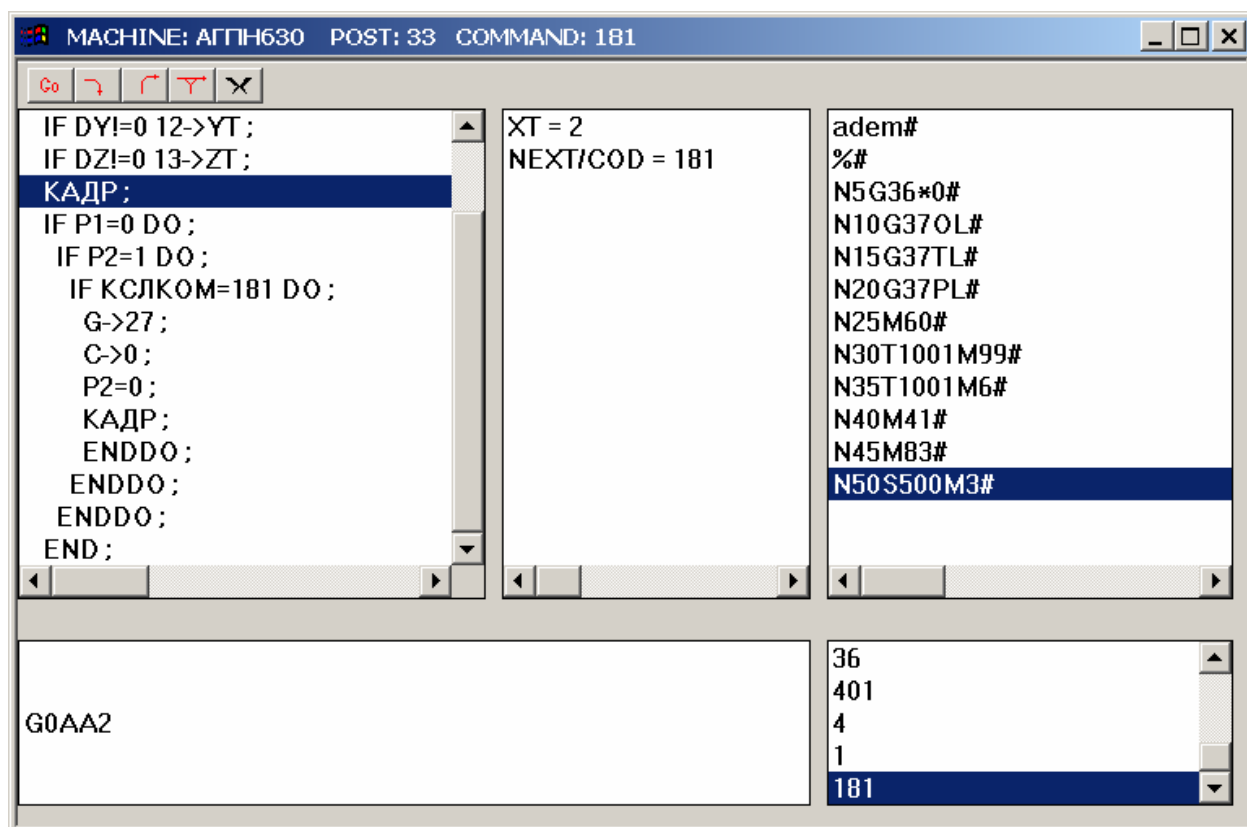


Рис. 19. Окно отладчика.

Информация на экране

В заголовке окна находится следующая информация:

- имя станка
- код обрабатываемой команды CLDATA
- номер постпроцессора

Левый верхний раздел окна:

- алгоритм на текущую команду CLDATA
(подсвечивается текущая строка алгоритма)

Центральный раздел окна:

- область задания системных и пользовательских переменных, используемых в алгоритмах для контроля их значений

Клавиша **INS** – позволяет ввести в поле имя переменной для контроля ее значения

Правый верхний раздел окна:

- формируемые кадры УП.

Левый нижний раздел окна:

- формируемый кадр УП.

Правый нижний раздел окна:

- список алгоритмов на команды CLDATA задействованный в постпроцессоре

(двойной щелчок мыши на номере алгоритма выводит его содержимое на экран).


Режимы работы отладчика

Отладчик может работать в двух режимах:

- *Полная отладка до точки останова(если задана)* .
- *Пошаговая отладка.*

Полная отладка выполняется при нажатии клавиши  (или клавиша – F8)

до точки останова (Breakpoint), которую отмечают двойным щелчком мыши на первом символе строки алгоритма (или клавиша – F9).

Пошаговая отладка выполняется при нажатии  клавиши -
(алгоритмы всех команд CLDATA выполняются *построчно*, точки останова игнорируются)

Дополнительные режимы отладки выполняются при

нажатии клавиш-



Первая клавиша дает возможность выйти из текущего алгоритма с выполнением его.

Вторая клавиша – обрабатывает вызовы алгоритмов (CALL) не входя в них.

Приложение А. Команды CLDATA

Наименование	Обозначение	Код
Программа	ПРОГРАМ	1
Деталь	ДЕТАЛЬ	2
Станок	СТАНОК	3
Конец УП	КОНЕЦ	4
Стоп программы	СТОП	22
Включить рабочую подачу	ПОДАЧА	23
Включить шпиндель	ШПИНД	24
Включить ускоренную подачу	ХХОД	25
Включить СОЖ	СОЖ	26
Включить выстой	ВЫСТОЙ	27
Отвести инструмент	ОТВОД	28
Перезахват (листоштамповка)	ПЕРЕХВАТ	29
Вспомогательная функция	ФУНВ	30
Подготовительная функция	ФУНП	31
Пропуск кадров	ПРОПУСК	32
Условный останов программы	ОСТАНОВ	33
Смена стола	СТОЛ	34
Загрузка инструмента	ИНСТР	35
Цикл	ЦИКЛ	36
Поворот стола	ПОВОРОТ	40
Совмещенное перемещение	МУЛЬТИ	41
Резьба	РЕЗЬБА	94
Линейная интерполяция	ИДИТОЧ	181
Круговая интерполяция	ИДИОКР	183
Перемещения с 3D-коррекцией	ИДИ ТОЧВКТ	189
Основное перемещение по пространственной кривой	ИДИ КРИВАЯ	190
Дополнительное перемещение по пространственной кривой	ИДИ ДОБАВОЧНОЕ КРИВАЯ	191
Вызов подпрограммы	ВЫЗОВ	223
Начало подпрограммы	ПОДПРОГ	252
Фиксация начала цикла	НЦ	401
Позиция смены инструмента	БЕЗПОЗ	451
Вставка в УП	ВСТАВКА	458

<i>Номер турели</i>	ТУРЕЛЬ	493
<i>Проверка условия</i>	ЕСЛИ	526
<i>Передача управления на метку</i>	ИДИМЕТ	527
<i>Метка</i>	МЕТКА	528
<i>Комментарий программы</i>	КОММЕНТ	582
<i>Выключить СОЖ</i>	СОЖ/ВЫКЛ	700
<i>Выключить шпиндель</i>	ШП/ВЫКЛ	701
<i>Завершить пропуск кадров</i>	ПР/ВЫКЛ	702
<i>Включить корректор по оси X</i>	KX/ВКЛ	703
<i>Включить корректор по оси Y</i>	KY/ВКЛ	704
<i>Включить корректор по оси Z</i>	KZ/ВКЛ	705
<i>Включить радиусный корректор</i>	KR/ВКЛ	706
<i>Выключить корректор по оси X</i>	KX/ВЫКЛ	707
<i>Выключить корректор по оси Y</i>	KY/ВЫКЛ	708
<i>Выключить корректор по оси Z</i>	KZ/ВЫКЛ	709
<i>Выключить радиусный корректор</i>	KR/ВЫКЛ	710
<i>Ориентированный останов шпинделя</i>	ШП/ОР	711
<i>Конец подпрограммы</i>		713
<i>Зарезервированная команда (ее алгоритм никогда не отрабатывается)</i>		799

Приложение Б. Пример паспорта станка

Постпроцессор, приведенный здесь в качестве примера, поставляется вместе с системой и имеет номер 222. Он подготовлен для станка модели 6313Ф3, стойка НЗЗ.

С. станок: Фрезерный.

2. Соответствие осей:

Хст->Хдет;

Yст->Yдет;

Zст->Zдет.

3. Максимальное перемещение:

по оси X - 0.0 мм;

по оси Y - 0.0 мм;

по оси Z - 0.0 мм.

4. Длина заправочной части ленты - 0.0 м.

5. Количество пропусков между кадрами: 3.

6. СОЖ.

Включение - вручную;

7. Подпрограммы - нет.

8. Интерполяция.

Линейная - нет;

Круговая - есть;

Максимальный радиус интерполяции 99999.0 м;

Перемещения по дуге разбиваются на октанты.

9. Аппроксимация по умолчанию - 0.009 м.

10. Инструмент.

Загрузка инструмента - ручная;

Загрузка первого инструмента в конце программы – загружать;

Координаты БЕЗПОЗ - не оговорено;

Плоскость смены инструмента - в точке НЦ;

Стандартная плоскость холостого хода - не формировать;

Плоскости перемещения на х/ходу по неск. координатам - XY, YZ, XZ, XYZ.

11. Подача.

Устанавливается - программно;

Диапазоны - бесступенчатое регулирование;

Место включения подачи - в одном кадре с перемещением.

12. Шпиндель.

Частота вращения шпинделя устанавливается - вручную;

Диапазоны - бесступенчатое регулирование;

Место включения шпинделя - немедленно после получения команды.

13. Коррекция по осям включается.

X - немедленно после получения команды;

Y - немедленно после получения команды;

Z - при первом перемещении по оси Z;
R - в одном кадре с перемещением.

14. Коррекция по осям выключается

X - немедленно после получения команды;
Y - немедленно после получения команды;
Z - при первом перемещении по оси Z;
R - в одном кадре с перемещением.

15. Начальный блок - %*

16. Конечный блок - *

17. Стандартный код формирования программы - ISO четный.

18. Станочные циклы - отсутствуют.

19. Изменяемые символы:

* -> 10;
W -> 136.

Приложение В. Пример файла макрокоманд

Постпроцессор, приведенный в данном руководстве в качестве примера, поставляется вместе с системой и имеет номер 222.

Макрокоманда = 35 (*Загрузка инструмента*)

KZ/ВЫКЛ	709	(<i>Выключить корректор по оси Z</i>)
ХХОД	25	(<i>Включить ускоренную подачу</i>)
ОТВОД	28	(<i>Отвести инструмент</i>)
ШП/ВЫКЛ	701	(<i>Выключить шпиндель</i>)
СОЖ/ВЫКЛ	700	(<i>Выключить СОЖ</i>)
ИНСТР	35	(<i>Загрузка инструмента</i>)

Макрокоманда = 4 (*Конец УП*)

KZ/ВЫКЛ	709	(<i>Выключить корректор по оси Z</i>)
ХХОД	25	(<i>Включить ускоренную подачу</i>)
ОТВОД	28	(<i>Отвести инструмент</i>)
ШП/ВЫКЛ	701	(<i>Выключить шпиндель</i>)
СОЖ/ВЫКЛ	700	(<i>Выключить СОЖ</i>)
КОНЕЦ	4	(<i>Конец УП</i>)

Приложение Г. Пример макета кадра

Постпроцессор, приведенный в данном руководстве в качестве примера, поставляется вместе с системой и имеет номер 222.

N[]G[]G[]X[]Y[]Z[]I[]J[]K[]F[]M[]L[]W____%[] []

N[]	НОМЕР КАДРА Макс. количество позиций под число – 3 Подавление левых – нет Интервал нумерации – 1
G[]	ПОДГОТОВИТЕЛЬНАЯ ФУНКЦИЯ Номер группы – 1 Макс. количество позиций под число – 2 Подавление левых – нет Признак модальности – в пределах кадра
G[]	ПОДГОТОВИТЕЛЬНАЯ ФУНКЦИЯ Номер группы – 2 Макс. количество позиций под число – 2 Подавление левых – нет Признак модальности – в пределах кадра
X[]	ЧИСЛОВАЯ ИНФОРМАЦИЯ Номер группы – 1 Макс. количество позиций под число – 6 Подавление левых – нет Вывод "+" для положительных чисел – есть Признак модальности – в пределах кадра Ось действия – X
Y[]	ЧИСЛОВАЯ ИНФОРМАЦИЯ Номер группы – 1 Макс. количество позиций под число – 6 Подавление левых – нет Вывод "+" для положительных чисел – есть Признак модальности – в пределах кадра Ось действия – Y
Z[]	ЧИСЛОВАЯ ИНФОРМАЦИЯ Номер группы – 1 Макс. количество позиций под число – 6 Подавление левых – нет Вывод "+" для положительных чисел – есть Признак модальности – в пределах кадра Ось действия – Z

I[]	<p>ЧИСЛОВАЯ ИНФОРМАЦИЯ</p> <p>Номер группы – 1</p> <p>Макс. количество позиций под число – 6</p> <p>Подавление левых – нет</p> <p>Вывод "+" для положительных чисел – есть</p> <p>Признак модальности – в пределах кадра</p> <p>Ось действия – X</p>
J[]	<p>ЧИСЛОВАЯ ИНФОРМАЦИЯ</p> <p>Номер группы – 1</p> <p>Макс. количество позиций под число – 6</p> <p>Подавление левых – нет</p> <p>Вывод "+" для положительных чисел – есть</p> <p>Признак модальности – в пределах кадра</p> <p>Ось действия – Y</p>
K[]	<p>ЧИСЛОВАЯ ИНФОРМАЦИЯ</p> <p>Номер группы – 1</p> <p>Макс. количество позиций под число – 6</p> <p>Подавление левых – нет</p> <p>Вывод "+" для положительных чисел – нет</p> <p>Признак модальности – в пределах кадра</p> <p>Ось действия – ?</p>
F[]	<p>ЧИСЛОВАЯ ИНФОРМАЦИЯ</p> <p>Номер группы – 2</p> <p>Макс. количество позиций под число – 4</p> <p>Подавление левых – нет</p> <p>Вывод "+" для положительных чисел – нет</p> <p>Признак модальности – в пределах кадра</p> <p>Ось действия – ?</p>
M[]	<p>ВСПОМОГАТЕЛЬНАЯ ФУНКЦИЯ</p> <p>Номер группы – 1</p> <p>Макс. количество позиций под число – 2</p> <p>Подавление левых – нет</p> <p>Признак модальности – в пределах кадра</p>
L[]	<p>ПОДГОТОВИТЕЛЬНАЯ ФУНКЦИЯ</p> <p>Номер группы – 2</p> <p>Макс. количество позиций под число – 3</p> <p>Подавление левых – нет</p> <p>Признак модальности – в пределах кадра</p>
W___%[]	СИМВОЛЬНОЕ ОКНО
][]	КОНЕЦ КАДРА.

Приложение Д. Пример файла алгоритмов

```
1; (Код команды CLDATA-Программа)
_Q=0;
_G17=0;
_G1=2;
_G2=3;
_X=4;
_Y=5;
_Z=6;
_I=7;
_J=8;
_K=9;
_F=10;
_M=11;
_L=12;
_P5=0;
END;
```

[illegible]

```
4;                                     (Конец УП)
_F->624;
КАДР;
_F->624;
_M->2;
АОКНО 13;
КАДР;
END;
```

```
22;          (Стоп программы)
КАДР;
_M->0;
КАДР;
END;
```

```

23;          (Включить рабочую подачу)
IF S>=1000 _F->4700+S*100;
ELSE IF S>=500 _F->4600+S*10;
ELSE IF S>=100 _F->600+S*10;
ELSE IF S>=10 _F->500+S;
ELSE IF S>=1 _F->400+S*10;
ELSE IF S>=0.1 _F->300+S*100;
ELSE _F->310;
_P1=0;
_P5=0;
END;

```



```
24;          (Включить шпиндель)
КАДР;
_M->3;
КАДР;
END;
```

```
25;          (Включить ускоренную подачу)
_P1=1;
END;
```

```
27;          (Включить выстой)
_G->4;
_X->ВЫСТВР;
_F->560;
END;
```

```
33;          (Условный останов программы)
_M->1;
END;
```

```
35;          (Загрузка инструмента)
IF _P1=1 DO;
_F->624;
КАДР;
_F->624;
_M->0;
КАДР;
ENDDO;
_P1=1;
END;
```

```
181;         (Линейная интерполяция)
IF КСЛКОМ=23 _P5=0 // СЛЕД. КОМАНДА - ВКЛ. ПОДАЧИ;
ELSE _P5=4000;
IF _P1=1 DO // УСКОРЕННОЕ ПЕРЕМЕЩЕНИЕ;
IF ДЛПЕР >400 _F->_P5+730;
ELSE IF ДЛПЕР >200 _F->_P5+724;
ELSE IF ДЛПЕР >10 _F->_P5+712;
ELSE _F->_P5+680;
ENDDO;
IF _Q=201 DO // ПОДХОД НА ВКЛ. РАДИУСНОЙ КОРРЕКЦИИ;
_L->800+_G17;
_Q=200;
ENDDO;
IF DX!=0 _X->DX;
IF DY!=0 _Y->DY;
IF DZ!=0 _Z->DZ;
IF _Q=400 DO // ВЫКЛЮЧЕНИЕ РАДИУСНОЙ КОРРЕКЦИИ;
_G1->50;
_L->_G17;
```

```

КАДР;
_G1→4;
_F→624;
_L→0;
_Q=0;
ENDDO;
ELSE IF _Q=300 DO // ВЫКЛЮЧЕНИЕ КОРРЕКТОРА ПО Z;
_G→40;
_L→400+_G20;
_Q=0;
ENDDO;
ELSE _G→1;
КАДР;
END;

```

```

183;          (Круговая интерполяция)
_X→DX;
_Y→DY;
IF ABS(XC-XЦОКР)>=0.01 _I→ABS(XC-XЦОКР);
IF ABS(YC-YЦОКР)>=0.01 _J→ABS(YC-YЦОКР);
IF НАПРОКР=ЧС _G→2;
ELSE _G→3;
IF _Q=200 DO // ВКЛЮЧЕНА РАДИУСНАЯ КОРРЕКЦИЯ;
IF БОКР=БОГН _L→800+_G17;
ELSE _L→_G17;
ENDDO;
КАДР;
END;

```

```

701;          (Выключить шпиндель)
КАДР;
_M→5;
END;

```

```

705;          (Включить корректор по оси X)
_G20=ВКЛКОРZ;
_L→ВКЛКОРZ+400;
END;

```

```

706;          (Включить радиусный корректор)
_G17=ВКЛКОРR;
_Q=201;
END;

```

```

710;          (Выключить радиусный корректор)
_Q=400;
END;

```

```

END;          (Конец алгоритма)

```