



Ductpost

Краткое

Справочное Руководство

Delcam plc, 1991-1994

Перевод на русский язык АО "ДЕЛКАМ-СПб", 1994

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ

- 1.1 ФАЙЛЫ ПОСТПРОЦЕССОРА
- 1.2 ФАЙЛЫ ДАННЫХ DUSTPOST

2 ЗАПУСК ПОСТПРОЦЕССОРА

- 2.1 СИНТАКСИС КОМАНДЫ
- 2.2 ИМЕНА ФАЙЛОВ ПО УМОЛЧАНИЮ.

3 ИЗМЕНЕНИЕ ЗНАЧЕНИЙ ПО УМОЛЧАНИЮ

- 3.1 ФАЙЛ STARTUP
- 3.2 ФАЙЛ СООБЩЕНИЙ

4 КОНФИГУРИРОВАНИЕ ПОСТПРОЦЕССОРА

- 4.1 ОСНОВНОЙ ФАЙЛ КОНФИГУРАЦИИ
- 4.2 ФОРМАТ СЛОВА
- 4.3 ЗАДАНИЕ МНОЖЕСТВЕННОГО ФОРМАТА
- 4.4 ПОРЯДОК СЛОВ
- 4.5 ОПРЕДЕЛЕНИЕ КЛЮЧЕЙ
- 4.6 ОПРЕДЕЛЕНИЕ КОДОВ
- 4.7 НАБОРЫ КАДРОВ ФИКСИРОВАННОЙ СТРУКТУРЫ
- 4.8 ПЕРЕМЕННЫЕ, ИСПОЛЬЗУЕМЫЕ В КОНФИГУРАЦИОННОМ ФАЙЛЕ
- 4.9 ФАЙЛ СООБЩЕНИЙ
- 4.10 КОНФИГУРАЦИОННЫЙ ФАЙЛ
- 4.11 ФАЙЛ MSD
- 4.12 CL-ФАЙЛ

5 ПОЛУЧЕНИЕ УПРАВЛЯЮЩЕЙ ПРОГРАММЫ

- 5.1 ЕДИНИЦЫ ИЗМЕРЕНИЯ
- 5.2 НАЧАЛО И КОНЕЦ УП
- 5.3 СТАРТОВАЯ ПОЗИЦИЯ
- 5.4 НОМЕРА КАДРОВ
- 5.5 ВЫВОД СООБЩЕНИЙ
- 5.6 ПРОМЕЖУТКИ
- 5.7 РАЗБИЕНИЕ УП НА НЕСКОЛЬКО ЧАСТЕЙ
- 5.8 ШПИНДЕЛЬ И ОХЛАЖДЕНИЕ
- 5.9 СМЕНА ИНСТРУМЕНТА
- 5.10 ЛИНЕЙНЫЕ ПЕРЕМЕЩЕНИЯ
- 5.11 КРУГОВАЯ ИНТЕРПОЛЯЦИЯ
- 5.12 БЫСТРЫЕ ПЕРЕМЕЩЕНИЯ
- 5.13 ЦИКЛЫ СВЕРЛЕНИЯ
- 5.14 ПЕРЕМЕЩЕНИЕ ГОНОМЕ

- ПРИЛОЖЕНИЕ А** СПИСОК КЛЮЧЕЙ
- ПРИЛОЖЕНИЕ Б** СПИСОК КОДОВ ФУНКЦИЙ

Соглашения, используемые в данном руководстве

Предполагается, что пользователь знаком с работой на компьютере и понимает значение таких терминов, как каталог, файл и команда.

- Строки, используемые как примеры ввода пользователя, будут выделяться специальным шрифтом.
- Вновь вводимые термины будут выделяться жирным шрифтом.
- Необязательные элементы заключаются в квадратные скобки [].
- В угловые скобки < > заключаются названия, которые в реальных командах должны быть заменены на конкретные объекты. Например, <имя файла> должно быть заменено на имя конкретного файла.
- Для термина "Управляющая программа для станка с числовым программным управлением" в тексте будет использоваться сокращение "УП".

Введение

DUCTpost может генерировать УП для любых станков с ЧПУ. Внося изменения в файлы данных (xxx.opt), можно настроить программу на вывод УП, подходящей для конкретного станка.

Команда на запуск DUCTpost выглядит так:

```
ductpost <имя станка> <имя CL-файла>
```

Пояснения относительно этой командной строки и CL-файла будут даны ниже.

Постпроцессор можно использовать только в том случае, если в файле защиты присутствуют следующие строки:

DUCTPOST-1.0-DUCTPOST

DUCTPOST-1.0-AGENT или DUCTPOST-1.0-CONFIG

а также строка для многокоординатной обработки

DUCTPOST-1.0-MULTAX

Постпроцессор читает Cldata файл, созданный в PowerMILL или DUCT и создает файл управляющей программы для конкретного станка (системы ЧПУ). Постпроцессор можно запускать двумя путями:

- Непосредственно из PowerMILL из Редактора УП,
- В командном окне. При этом cut- файл должен уже существовать на диске.

Библиотека постпроцессоров содержит около 50 стандартных постпроцессоров. Модификация формата УП осуществляется с помощью файла опций. Файл опций – текстовый файл, использующий тот же формат, что и стандартные постпроцессоры. С его помощью можно настроить и новый постпроцессор.

Файлы постпроцессора

В процессе выполнения постпроцессор считывает CL-файл, а после обработки данных выводит три файла: файл с управляющей программой, файл с комментариями и информацией об инструменте и файл вывода на печать, который содержит подробную информацию об управляющей. Все эти файлы записываются в текущий каталог.

Файлы данных DUCTpost

В процессе работы DUCTpost использует следующие файлы:

1. Файл с сообщениями dcam/ductpostxxx/sys/hci/ductpost.msg. Этот файл содержит тексты всех сообщений, генерируемых постпроцессором.
2. Файл dcam/ductpost1007/sys/misc/startup. Этот файл содержит имена по умолчанию всех файлов, считываемых и записываемых постпроцессором.
3. Конфигурационный файл dcam/config/ductpost/.....cfg. Этот файл создается для каждого станка и содержит информацию об особенностях станка и системы ЧПУ. Это обычный текстовый файл, его легко модифицировать. Его первая строка должна содержать *machine.....*
а последняя
end
4. Файл опций /.....opt. Этот файл содержит небольшие отличия между станками. Для каждого конфигурационного файла можно использовать несколько файлов опций. Конфигурационный файл содержит информацию об основных особенностях системы ЧПУ, а файлы опций содержат информацию о небольших отличиях (например, координаты абсолютные или в приращениях, заголовок перфоленты и т.д.). Формат файла опций совпадает с форматом конфигурационного файла. Первая строка файла опций должна содержать имя конфигурационного файла.

Поиск файлов выполняется сначала в текущем каталоге, а затем уже в библиотечном каталоге dcam/ductpostxxx/lib.

Запуск постпроцессора

Синтаксис команды

Формат команды для запуска постпроцессора:

```
ductpost [ключи] [<имя файла опций>] [<имя CL-файла>]
```

Например, команда

```
ductpost fanuc6m test
```

запускает DUCTpost для CL-файла test.cut с выводом УП для станка с системой ЧПУ FANUC6M.

Имена файлов в команде надо задавать в указанном порядке. Если задан только один аргумент, он рассматривается как имя CL-файла, тогда имя файла опций будет запрошено при выполнении постпроцессора. Если ни одно имя файла не задано в команде, при выполнении будут запрошены оба имени. В команде на запуск DUCTpost можно задавать следующие ключи:

- s <имя файла> файл с указанным именем будет использоваться в качестве файла STARTUP вместо файла startup.
- m <имя файла> файл с указанным именем будет использоваться в качестве файла сообщений вместо файла ductpost.msg.
- p <имя файла> файл с указанным именем будет использоваться в качестве файла вывода на печать. Если вместо имени файла задано *none*, вывод подавляется, а если задано *tty*, используется стандартный вывод.
- t <имя файла> файл с указанным именем будет использоваться в качестве файла для вывода УП. *none* и *tty* используются так же, как и для файла печати.

Например, по команде

```
ductpost -t tty -p none fanuc6m test
```

вводится CL-файл test.cut, файл печати не создается, а файл с УП выводится на терминал.

По команде

```
ductpost -p none fanuc6m  
Enter job file name : xxx
```

вводится CL-файл xxx.cut, выводятся файлы xxx.tap и xxx.inf, а файл печати не создается.

Имена файлов по умолчанию

Имена файлов, задаваемые в команде с помощью ключей, имеют следующие значения по умолчанию:

Файл STARTUP.	startup
Файл сообщений.	ductpost.msg
Файл с УП.	Имя CL-файла с расширением .tap
Файл печати.	Имя CL-файла с расширением .prt
Информационный файл	Имя CL-файла с расширением .inf

Изменение значений по умолчанию

Файл STARTUP

Файл STARTUP не является обязательным. Если он присутствует, он содержит имена и расширения по умолчанию CL-файла, файла УП и файла печати. Порядок строк в этом файле не важен, и перечисление всех файлов не является обязательным.

Формат строки этого файла следующий:

<файл> <часть> = <имя>

<файл> имеет одно из значений 'cldata', 'tape' или 'print'. *<часть>* может быть либо 'file', либо 'extension'. *<имя>* может принимать значения 'tty' или 'none', которые имеют те же значения, что и в командной строке. Типичная строка файла может выглядеть так:

print file = none

Имя файла сообщений задается следующим образом:

message file = <filename>

Расширение имени файла сообщений задать нельзя, и 'tty' или 'none' также не имеют никакого смысла. Файл опций задается вместе с заданием системы ЧПУ. Форма команды:

machine <filename>

Пример содержимого файла

```
STARTUP: print file = none
tape extension = .t01
machine hurco
```

Здесь задается расширение .t01 вместо .tap для файла с УП, подавление файла печати и имя файла опций hurco.opt.

Файл сообщений

Файл сообщений содержит почти весь текст, который выводится постпроцессором. Каждая строка этого файла содержит сообщение или часть сообщения в форме

```
<message index> "<message text>"
```

<message index> - это число от 1 до максимального номера сообщения (обычно это 200). Текст представляет из себя строку в кавычках. Если первым символом строки является плюс, она является продолжением предыдущей. Количество строк продолжения не ограничено, но общее количество строк сообщения не должно превышать 240.

Индекс сообщения является фиксированным для каждого сообщения. Например, сообщение 11 всегда выводит заголовок файла. Если к базе данных добавить новые сообщения, они не будут выводиться программой. Но если внести изменения, программа будет их использовать.

Конфигурирование постпроцессора

Файл опций определяет, каким образом из CL-файла генерируется УП. Файл опций состоит из двух секций. В первой описываются функции станка и соответствующие этим функциям кадры УП. Во второй описываются переменные, влияющие на выполнение постпроцессора.

Основной файл конфигурации

Файл конфигурации является текстовым файлом в формате ASCII, его можно редактировать обычным текстовым редактором. Первая строка файла содержит имя станка, а последняя - слово 'end'. Простейший файл конфигурации выглядит следующим образом

```
machine f11m  
end
```

Файл опций (если он существует) имеет точно такой же формат. В файл конфигурации можно добавлять комментарии. Строка с комментарием должна начинаться с символа '#'. Такая строка будет игнорироваться постпроцессором, также, как и пустая строка. Строчные и прописные буквы различаются постпроцессором, поэтому 'F11M' не эквивалентно 'f11m'.

Данные в конфигурационном файле разделяются на следующие категории:

1. Лексемы. Лексемой называется любой набор символов, ограниченный пробелами, например, 'f11m', 'lab1', '+-*='.
2. Ключевые слова. Ключевым словом называется последовательность лексем, которая имеет специальное значение. Ключевые слова могут содержать более одной лексемы, при этом количество пробелов между лексемами не имеет значения. Каждая лексема может иметь аббревиатуру, которая должна быть уникальной. Например, ключевое слово 'machine' может быть сокращено до 'mach', но не 'mac'.
3. Целые константы. Целая константа состоит из набора цифр, которому предшествует необязательный знак плюс или минус, например, '27', '+012345', '-45678'.
4. Вещественные константы. Вещественная константа состоит из набора цифр, которому предшествует необязательный знак плюс или минус, а также десятичной точки, например, '27.0', '+012345.', '-45678.9'.
5. Текстовые строки. Текстовой строкой является любой набор символов, заключенный в двойные кавычки, например, "This is a string". Пробелы внутри текстовой строки являются значащими, а в конце - нет. Например, две текстовые строки "This is a string" и "This is a string " являются идентичными, а строка "This is a string" отлична от них обеих.
6. Списки. Списком называется набор лексем, разделенных пробелами и заключенных в круглые скобки. Например, '(this is a list)' является списком, содержащим лексемы 'this', 'is', 'a' и 'list'.
7. Массивы. Массивом называется список целых чисел, вещественных чисел или текстовых строк.

Формат слова

Управляющая программа состоит из кадров, которые содержат слова. Каждое слово состоит из адреса, за которым следует числовое значение. Типичный кадр выглядит так:

GO Y-40. X-25. S800 M3

Каждому слову соответствует описание его формата, включающее букву адреса и форму вывода числового значения. Общая форма описания формата слова:

*define word <метка>
<описание формата>
end define*

*определить слово
конец определения*

Пример описания формата слова X_:

<i>define word X_</i>	
<i>address letter = "X"</i>	<i>буква адреса "X"</i>
<i>adress width = 1</i>	<i>ширина адреса 1 символ</i>
<i>field width = 8</i>	<i>ширина поля числового значения 8</i>
<i>tape position = 1</i>	<i>позиция при выводе в УП 1</i>
<i>print position = 0</i>	<i>позиция при выводе на печать 0</i>
<i>scale factor = 2</i>	<i>масштаб 2</i>
<i>sign = if negative</i>	<i>перед числом выводить только минус</i>
<i>not modal</i>	<i>не модальное</i>
<i>not permanent</i>	<i>не постоянное</i>
<i>metric formats</i>	<i>метрические единицы измерения decimal</i>
<i>places = 3</i>	<i>3 десятичных знака после запятой</i>
<i>decimal point = true</i>	<i>десятичную точку выводить</i>
<i>leading zeros = false</i>	<i>предшествующие нули не</i>
<i>выводить trailing zeros = true</i>	<i>последующие нули выводить</i>
<i>imperial formats</i>	<i>дюймовые единицы измерения</i>
<i>decimal places = 4</i>	<i>4 десятичных знака после запятой</i>
<i>decimal point = true</i>	<i>десятичную точку выводить</i>
<i>leading zeros = false</i>	<i>предшествующие нули не выводить</i>
<i>trailing zeros = true</i>	<i>последующие нули выводить</i>
<i>end define</i>	

Данное определение означает, что в качестве адреса используется буква X, за которой следует число из восьми цифр с десятичной точкой и тремя десятичными знаками после точки. Перед числом выводится только знак минус в случае, если число отрицательное. Предшествующие нули не выводятся, а последующие выводятся, например,

X45.234 или X230.100

Слово не является модальным. После слова в УП выводится один пробел, а в файл печати пробел не выводится. После буквы адреса пробел также не выводится, поскольку ширина адреса задана равной 1. Числовое значение будет увеличиваться согласно заданному масштабу в два раза. Это иногда используется в токарных станках, где координата X должна представлять диаметр, а не радиус. Формат вывода в метрических единицах отличается от формата вывода в дюймовых единицах только количеством знаков после запятой.

Пример описания формата слова G_:

<i>define word G_</i>	
<i>address letter = "G "</i>	<i>буква адреса "G"</i>
<i>adress width = 2</i>	<i>ширина адреса 2 символа</i>
<i>field width = 2</i>	<i>ширина поля числового значения 2</i>
<i>tape position = -5</i>	<i>позиция при выводе в УП -5</i>
<i>print position = -5</i>	<i>позиция при выводе на печать -5</i>
<i>scale factor = 1</i>	<i>масштаб 1</i>
<i>sign = none</i>	<i>знак не выводится</i>
<i>modal</i>	<i>модальное</i>
<i>not permanent</i>	<i>не постоянное</i>
<i>metric formats</i>	<i>метрические единицы измерения</i>
<i>decimal places = 0</i>	<i>целое число</i>
<i>decimal point = false</i>	<i>десятичную точку не выводить</i>
<i>leading zeros = true</i>	<i>предшествующие нули выводить</i>
<i>trailing zeros = true</i>	<i>последующие нули выводить</i>
<i>imperial formats = metric formats</i>	<i>метрический формат</i>
	<i>совпадает с дюймовым</i>
<i>end define</i>	

Данное определение означает, что в качестве адреса используется буква G, слово является модальным, а число является целым и не имеет десятичной точки, например,

G02 или G10

Слово всегда будет появляться в шестой позиции от начала кадра как в файле УП, так и в файле печати.

Если несколько слов имеют одинаковые форматы, их определения можно объединить, например,

```
define word X_
  address letter = "X"
end define
define word Y_
  address letter = "Y"
end define
define word Z_
  address letter = "Z"
end define
define format (X_ Y_ Z_)
  field width = 8 scale
  factor = 1
  . . . . .
end define
```

Каждой команде **define word** должна соответствовать команда **end define**. Каждое определение, начинающееся с **define**, должно обязательно заканчиваться **end define**.

Метка, которой обозначается слово в команде **define word**, используется в дальнейшем в конфигурационном файле для ссылки на это определение. В качестве метки можно использовать любую

лексему кроме тех, которые распознаются постпроцессором в качестве ключевых слов или их аббревиатур. Значащими являются первые 12 символов метки.

Описание формата слова состоит из определения переменных в форме *<variable> = <value>*

по одному определению в строке.

Ниже приводится список всех переменных, используемых для описания формата слова.

adress letter	Строка символов, используемая в качестве адреса слова, может иметь длину до 20 символов.
adress width	Целое число, количество позиций, которые занимает адрес в УП. Если это число превосходит количество букв адреса, в УП будут выведены дополнительные пробелы справа. Если же оно меньше количества букв адреса, адрес будет усечен.
field width	Целое число, задающее количество позиций, которое отводится под числовое значение.
tape position	Целое число, указывающее на позицию данного слова относительно других слов в кадре при выводе УП. Положительное число указывает на смещение данного слова относительно предыдущего слова в кадре. Отрицательное число указывает на смещение данного слова относительно начала кадра.
print position	Целое число, указывающее на позицию данного слова относительно других слов в кадре при выводе на печать. Аналогично предыдущему.
scale factor	Целое число, коэффициент масштабирования числового значения.
sign	Управляет выводом знака перед числом. Может принимать значения: none - не выводить, if negative - выводить только минус, always - выводить всегда.
modality	Модальность. Слово называется модальным, если оно выводится только при изменении его числового значения, в этом случае в строке задается слово modal . В противном случае задается слово not modal .
permanence	Постоянный вывод. Слово называется постоянным, если оно должно выводиться в каждом кадре, в этом случае в строке выводится permanent . В противном случае выводится not permanent .
units specifier	Числовые величины могут выводиться в УП как в метрической

системе, так и в дюймовой, при этом формат вывода может быть различным. Утверждение **metric format** означает, что последующие определения формата будут применяться к величинам в метрической системе, а утверждение **imperial format** означает, что последующие определения формата будут применяться к величинам в дюймовой системе.

decimal places	Целое число, определяющее количество десятичных знаков после запятой при выводе числа. Ноль означает, что выводится целое число.
decimal point	Управление выводом десятичной точки. Может принимать значения : true - выводить, false - не выводить. Например, при выводе числа 1 с тремя знаками после запятой без десятичной точки получится 1000, а с десятичной точкой получится 1.000.
leading zeros	Управление выводом предшествующих нулей. Может принимать значения : true - выводить, false - не выводить. Например, при выводе числа 1 в поле из 5 символов с предшествующими нулями получится 00001, а без них - 1.
trailing zeros	Управление выводом последующих нулей. Может принимать значения : true - выводить, false – не выводить. Например, при выводе числа 1 с десятичной точкой и тремя знаками после запятой с последующими нулями получится 1.000, а без них - 1. Для целых чисел последующие нули всегда выводятся, иначе вместо 1000 получится 1!
format copy	Когда метрический и дюймовый форматы совпадают, их можно скопировать один в другой.

Задание множественного формата

При создании конфигурационного файла для системы ЧПУ с большим количеством слов форматы многих из них могут совпадать. В этом случае удобно использовать описание формата для нескольких слов одной командой. Для этого используется команда **define format** в форме

<i>define format</i> <список>	<i>определить формат</i>
<описание формата>	
<i>end define</i>	<i>конец определения</i>

<список> можно заменить на слово **all** или на список ранее определенных меток слов. Здесь можно применять любое описание

формата из приведенных выше. Важно отметить, что максимальная длина строки в конфигурационном файле составляет 128 символов.

Пример 1

При задании формата координат указывается: ширина поля для числа, ставить или не ставить десятичную точку, число знаков после запятой, ставить ли знак числа и выводить или нет предшествующие и последующие нули. Все это можно делать для каждой координаты отдельно.

```
define format (X Y Z I J K )
  field width = 8
  sign = if negative
  modal
  decimal places = 3
  decimal point = true
  leading zeros = false
  trailing zeros = false
end define
```

Пример 2

Задание абсолютных координат (иначе указывается incremental):
coordinates = absolute

Формат G- функций. Чтобы, к примеру, линейная интерполяция задавалась G01, а не G1, надо задать ширину поля 2 и разрешить вывод предшествующих нулей (смотри ниже). Параметр modal задает модальность, т.е. функцию выводить не надо, если она не изменилась.

```
define format (G1 G2 G3 G5 G7 )
  field width = 2
  leading zeros = true
  modal
end define
```

Порядок слов

Каждое слово может появляться в кадре только один раз, при этом порядок слов в кадре является строго определенным. Он задается переменной **word order**. Слова в кадре будут появляться в том порядке, в каком они перечислены в этой переменной.

Поскольку строка определения может быть очень длинной, последующее определение со знаком + означает, что этот список добавляется к предыдущему.

Например,

```
word order = (N_ G_ S_)
word order = (+ X_ Y_ Z_ I_ J_ K_ F_ M_)
word order = (+ msg_beg msg msg_end)
```

означает, что слова будут выводиться в кадре в

следующем порядке:

```
N_ G_ S_ X_ Y_ Z_ I_ J_ K_ F_ M_ msg_beg msg msg_end
```

Все слова надо обязательно включить в этот список, хотя они могут и не выводиться в каждом кадре. Если слово отсутствует в списке, оно вообще не будет выводиться в УП.

msg используется для определения позиции текста, содержащегося в CL-записи PPRINT.

Слово можно включать в список только после определения его формата, поэтому порядок слов устанавливается только после того, когда все слова уже определены.

Определение ключей

Ключом называется элемент, который может менять свое значение от кадра к кадру. Примерами ключей являются номера кадров и значения координат X, Y и Z.

Каждому ключу должно соответствовать слово, определяющее формат вывода ключа. Это соответствие назначается в секции определения ключей в следующей форме:

```
define keys                                определить ключи
    <имя ключа> = <метка формата слова>
end define                                  конец определения
```

Пример

```
define keys                                определить ключи
    blocknumber = N_                        номер кадра
    x coordinate = X_                       координата X
    y coordinate = Y_                       координата Y
    z coordinate = Z_                       координата Z
    spindle = S_                            шпиндель
    auxfun = M_                             вспомогат. функция
    feedrate = F_                           подача
end define                                  конец определения
```

Количество определений ключей не ограничено. Одна строка должна содержать одно определение. Полный список ключей приведен в **Приложении А**. Метка формата слова должна относиться к ранее определенному слову.

Определение кодов

Код является предварительно определенным элементом, значение которого не меняется. Коды используются для вывода в УП кодов функций, которые обычно имеют стандартные адреса G и M. Каждый код связан с определенным словом и числовым значением, например, G0 задает ускоренную подачу. Секция определения кодов сходна с секцией определения ключей:

```
define codes                                определить коды
  <имя функции> = <метка формата слова><значение кода
  функции> end define                       конец
определения
```

Пример

```
define codes                                определить коды
  rapid = G1 0                               быстрый ход
  linear = G1 1                              линейное перемещение
  dwell = G2 4                               задержка
  abs data = G3 90                           абсолютные координаты
  incr data = G3 91                          координаты в приращениях
end define                                    конец определения
```

В этом примере предполагается, что слова с метками G1, G2 и G3 уже были определены. Несколько кодов с различными значениями, связанных с одним словом, определяют модальную группу.

Количество определений кодов не ограничено. Одна строка должна содержать одно определение. Полный список кодов приведен в **Приложении Б**.

Если постпроцессор попытается вывести ключ или код, которые не связаны ни с каким определением формата слова, будет выдано сообщение об ошибке. В том случае, если система ЧПУ не поддерживает какую-либо функцию, а постпроцессор ее выводит, коду функции следует присвоить значение -1.

Наборы кадров фиксированной структуры

Хотя описанные выше возможности обеспечивают большую гибкость при генерации УП, иногда возникает необходимость использовать наборы кадров фиксированной структуры, например

- в начале и конце УП
 - при смене инструмента
 - при отработке команды GO HOME
 - при выводе CL-записи типа 5000
- и т.д.

Определение такого набора состоит из списка спецификаций для выводимых слов через точку с запятой:

<i>define block <имя блока></i>	<i>определить блок</i>
<i><список слов></i>	
<i>end define</i>	<i>конец определения</i>

Пример

<i>define block tape start</i>	<i>определить кадры начала УП</i>
<i> "%"</i>	
<i> OAID_ProgID</i>	
<i> N_=P ; abs data ; G2 17 ; G3 40 ; G4 80</i>	
<i> N_=P ; X_FromX ; Y_FromY</i>	
<i>end define</i>	<i>конец определения</i>

В результате начало УП будет выглядеть следующим образом:

```
%
O1234
N010G90G17G40G80
N020X. . .Y. . .
```

Здесь N_=P означает, что номер будет выведен в кадре только в том случае, если в определении формата слова с меткой N_ было задано **permanent** (постоянное). Слова с метками G2, G3 и G4 имеют адрес G и поле из двух символов для задания числового значения, а код *abs data* определен как функция G с кодом 90.

Спецификация может иметь следующий вид:

1. Строка. В этом случае строка вставляется в кадр без изменений, а ее позиция определяется позицией **msg** в переменной **word order**.
2. Имя кода функции. Код функции будет выведен в соответствии с информацией из секции определения кодов функций.
3. Имя ключа или метка слова, за которым следует некоторое значение. В качестве значения может использоваться:
 - а) Целое число. Слово (или слово, соответствующее ключу) будет выводиться с этой величиной. Если слово имеет вещественный формат, заданное число надо промасштабировать, чтобы оно выводилось корректно. Например, если формат слова имеет три знака после запятой, число 1000 будет выведено как 1.000.
 - б) Строка символов. Строка символов будет выводиться вслед за буквой адреса. Она будет усечена справа или, наоборот, дополнена пробелами, в зависимости от длины поля значения в формате слова.
 - в) "=" . Слово или ключ выводятся в соответствии со своим текущим значением. г)
 - "=S". Слово выводится вместе со своим текущим значением только

ТОМ

случае, если его значение изменилось с момента его предыдущего вывода в кадр.

- д) "=С". Слово выводится вместе со своим текущим значением даже в том случае, если оно является модальным, а его значение не изменилось с момента его предыдущего вывода в кадр.
- е) "=Р". Слово выводится вместе со своим текущим значением только в том случае, если оно определено как **permanent** (постоянное).
- ж) Программная переменная. При формировании кадров могут использоваться текущие значения переменных. Список таких переменных представлен ниже.

Имя переменной Значение

BlockNumber	Текущий номер кадра.
ClearPlane	Плоскость безопасности для циклов сверления.
ClearPlaneInc	Расстояние между текущим значением Z и плоскостью безопасности для циклов сверления.
Counter	Счетчик.
Day	День текущей даты (формат nn).
FeedRate	Текущая подача.
FromX	Координата X стартовой точки обработки.
FromY	Координата Y стартовой точки обработки.
FromZ	Координата Z стартовой точки обработки.
HoleDepth сверления.	Глубина отверстия для циклов сверления.
HoleDepthInc	Приращение глубины отверстия.
Jobname	Имя CL-файла.
Month	Месяц текущей даты.
NextTool	Номер следующего инструмента.
NumberPecks сверления.	Количество углублений для циклов сверления.
Old X	Предыдущее значение координаты X.
Old Y	Предыдущее значение координаты Y.
Old Z	Предыдущее значение координаты Z.

PartID	Идентификатор программы. Может быть взят из CL-записи PARTNO, из конфигурационного файла или введен с клавиатуры в ответ на запрос.
PeckDepth	Высота одного углубления для циклов сверления.
ProgID	Номер программы. Увеличивается на 1 для каждого сегмента УП. Начальное значение может быть взято из конфигурационного файла или введено с клавиатуры в ответ на запрос.
SafeZ	Высота плоскости безопасности.
SpindleSpeed	Текущая скорость вращения шпинделя.
Tape Coord	Код функции текущего состояния координат (абсолютные или в приращениях).
TapeUnits	Код функции текущих единиц измерения (метрические или дюймовые).
TipRadius	Радиус режущей кромки текущего инструмента.
ToolFeed	Первая подача после смены инструмента.
ToolLenOff	Номер корректора длины инструмента.
ToolLength	Длина текущего инструмента.
ToolNumber	Номер текущего инструмента.
ToolRadOff	Номер корректора радиуса инструмента.
ToolRadius	Радиус текущего инструмента.
ToolSpeed	Скорость вращения шпинделя текущего инструмента.
Year	Год текущей даты.

Пример полного определения фиксированного набора кадров:

```

define block move rapid                                опред. кадров быстрого хода
  blockn BlockNum ; rapid ; spindle =S ;
  grp_12 =S ; grp_14 =S ; tool length
  ToolLen ; tool length =S ; x coord =S ; y
  coord =S ; z coord =S ;
end define                                             конец определения

```

Во время вывода фиксированного набора кадров выводятся только те слова, которые перечислены в определении. Если слово является постоянным, но отсутствует в определении, оно не будет выведено. Для кодов будут использованы не текущие значения, а значения, которые были установлены перед определением данного фиксированного набора кадров.

Ниже приведен полный список кадров и их место в УП.

tape start	Начало УП. Выводится в начале каждого сегмента УП.
tape end	Конец УП. Выводится в конце каждого сегмента УП.
go home preamble	Выводится в начале отработки CL-записи GO HOME. Этот набор кадров выводится даже в том случае, если флаг GO HOME сброшен.
go home xymove	Выводится при отработке CL-записи GO HOME, если меняются координаты X и Y.
go home zmove	Выводится при отработке CL-записи GO HOME, если меняется координата Z.
tool change first	Если этот набор кадров задан, он используется при установке первого инструмента.
tool change	Используется при смене инструмента, а если предыдущий набор не задан, то и для установка первого инструмента.
tool change clear	Если этот набор определен, значения всех модальных слов обнуляются, а затем выводится данный набор кадров.
move circle	Используется для вывода координат XYZ при отработке CL-записи типа 5000, соответствующей круговой интерполяции.
move rapid	Используется для вывода координат XYZ при отработке CL-записи типа 5000, соответствующей быстрому перемещению.
move linear	Используется для вывода координат XYZ при отработке CL-записи типа 5000, соответствующей линейной интерполяции.
move from	Используется для вывода координат XYZ при отработке CL-записи типа 5000, соответствующей заданию стартовой точки обработки.
move cycle	Используется для вывода координат XYZ при отработке CL-записи типа 5000 в стандартном цикле.

Все эти наборы кадров должны быть определены, иначе при отработке постпроцессором CL-записи типа 5000 будет выдано сообщение об ошибке.

Можно также определить набор кадров, которые будут выводиться при отработке CL-записей типа 2000. Общая форма такого определения:

```
define block cldat <N>  
  <список слов>  
end define
```

Здесь <N> представляет подтип записи типа 2000. Например, можно определить набор кадров для записи подтипа 1044 следующим образом:

```
define block cldat 1044
    msg_start 0 ; msg " " ;
msg_end 0 end define
```

Это определение даст эффект подавления номера кадра при выводе сообщения. Любая строка, следующая за **msg**, будет игнорироваться и заменяться на текст, содержащийся в CL-записи.

Переменные, используемые в конфигурационном файле

В конфигурационном файле можно задавать различные переменные. Они могут использоваться для хранения значений, используемых постпроцессором, а также для управления выводом результатов работы постпроцессора. Общая форма задания переменной:

<имя переменной> = <значение>

Тип переменной должен соответствовать типу значения, в противном случае будет выдано сообщение об ошибке, и присвоение значения не произойдет. Можно использовать переменные следующих типов:

1. Вещественные.
2. Целые.
3. Логические.
4. Текстовые.
5. Массивы.

Вещественные переменные

diameter	См. параметры многокоординатной обработки.
maximum feedrate	Максимально допустимая скорость подачи. Если задана большая скорость, она будет уменьшена до этого значения.
maximum tape length	Максимальная длина одной ленты. Если задать 0, длина ленты не ограничена.
minimum feedrate	Минимально допустимая скорость подачи.
rapid feedrate	Величина ускоренной подачи, используется для подсчета времени обработки.
withdrawal amount	См. параметры многокоординатной обработки.

Целые переменные

block increment	Приращение номера кадра. Номер кадра увеличивается только в том случае, если слово
------------------------	--

block start	с номером кадра выводится в УП.
program id start	Номер первого кадра. Начальное значение номера программы. Увеличивается на 1 для каждого нового сегмента УП.
split move	Разделение перемещения по осям на несколько кадров. 0 - координаты по всем осям выводятся в одном кадре 1 - перемещение по Z откладывается до следующего быстрого перемещения.
tape headers	Определяет содержание заголовка программы. 0 - нет заголовка 1 - номер программы и дата 2 - только номер программы 3 - только дата.
tool reset coordinates	Определяет вывод координат при смене инструмента. 0 - никаких специальных действий 1 - вывод координат X и Y вслед за сменой инструмента 2 - вывод координат X, Y и Z вслед за сменой инструмента.

Логические переменные

go home output	Разрешение на вывод перемещения GONOME.
message output	Разрешение на вывод сообщений из CL-файла в файл УП.
tlo output	Разрешение на вывод кода функции Tool length offset при смене инструмента.
use partid	Разрешение на использование идентификатора программы из CL-файла. В качестве альтернативы используется переменная PartId .
use progid	Разрешение на использование заданного номера программы. В качестве альтернативы используется ввод номера программы с клавиатуры.

Текстовые переменные

machine name	Имя станка, печатается в начале работы постпроцессора.
point	Символ, используемый в качестве десятичной точки.
print header	Строка, используемая в качестве заголовка файла при печати.
zero	Символ, используемый для вывода вещественного числа, равного нулю.
partid	Значение переменной PartId .

Массивы

circle output	Целый массив из 4 элементов, управляет выводом круговой интерполяции.
----------------------	---

1 - 1 = вывод окружности,
1 - 0 = нет вывода окружности
2 - 0 = предварительный вывод параметров
2 - 1 = вывод с перемещением по XY
2 - 3 = вывод только перемещений
3 - разбиение окружности на квадранты, 1 - разбивать
4 - 1 в приращениях / 0 - в абсолютных координатах.

coolant output

целый массив из 2 элементов, управляет выводом включения и выключения охлаждения.

1 - включение

2 - выключение.

Каждый элемент массива принимает значения

0 = выводится, если встречена CL-запись

1 = вывод в следующем кадре

2= вывод со следующим перемещением.

compensation output

Целый массив из 3 элементов, управляет выводом компенсации инструмента.

1 - вывод компенсации (0 = нет вывода)

2 - вывод смещения на радиус (0 = нет вывода)

3 - 1 = выключение компенсации в отдельном кадре

3- 2 = включение компенсации в отдельном кадре.

spindle output

Целый массив из 4 элементов, управляет выводом включения и выключения шпинделя.

1 - включение

2- 1 = вывод S0 для выключения шпинделя

3 - 1 = вывод кода функции включения /
выключения шпинделя.

tr

Текстовый массив из двух элементов, используется для

изменения символов при выводе сообщений.

Каждый символ первой строки будет заменен на соответствующий символ второй строки.

Файл сообщений

Поскольку этот файл поставляется вместе с постпроцессором, его использование обычно не вызывает появления сообщений об ошибках.

Тем не менее при чтении этого файла могут быть выданы следующие предупреждения:

The database overflows. (База данных переполнена.) В настоящий момент в базе может находиться не более 200 сообщений длиной до 240 строк каждое.

No valid message number is found. (Номер сообщения не найден.) Первым элементом каждой строки может быть либо номер от 1 до 200, либо знак "+".

A message has been duplicated. (Сообщение продублировано.) Это сообщение появляется в том случае, если какой-либо номер повторяется.

Если при чтении файла сообщений встречается ошибка, постпроцессор прекращает свою работу. Могут встречаться также ошибки при выводе сообщений на печать. Причиной таких ошибок может быть не заданное сообщение или неверный формат. В этом случае постпроцессор выдаст сообщение об ошибке и продолжит работу.

Конфигурационный файл

Конфигурационный файл является основным источником ошибок при выполнении конфигурируемого постпроцессора. Проблемы при чтении этого файла вызовут вывод предупреждения. Ошибки при чтении конфигурационного файла могут возникнуть в том случае, если постпроцессор не сможет понять смысл каких-либо утверждений, например,

```
cldata fule = job.cut
```

Здесь вместо слова *file* ошибочно использовано слово *fule*. Подобная ошибка вызовет сообщение об ошибке синтаксиса в слове *fule*. Если же ошибка встречена при чтении ключевого слова, сообщение будет относиться к первой лексеме ключевого слова, даже если ошибка находится в другом месте.

Ошибочным является также присвоение переменной значения другого

```
tape headers = " yes"
```

поскольку *tape headers* является целой переменной.

Неверным будет и использование переменной вне соответствующей секции, например, присвоение

```
decimal places = true
```

можно выполнять только в секции определения формата слов.

Ошибка в утверждении *define* вызовет большое количество сообщений об ошибках в последующих строках, поскольку они не могут восприниматься правильно без команды *define*, например,

```
define formattt (blkno  
  prefun) not modal  
  trailing zeros = true  
end define
```

вызовет вывод четырех сообщений об ошибках, хотя имеется только одна.

Любая ошибка при чтении конфигурационного файла вызовет вывод соответствующего сообщения и прекращение работы постпроцессора.

Файл MCD

Поскольку файл MCD генерируется постпроцессором, ошибки могут возникнуть только в том случае, если файл испорчен. Если постпроцессор не сможет прочитать этот файл, он прекратит свою работу.

CL-файл

Предполагается, что CL-файл не содержит ошибок и его записи имеют определенный формат. Ошибки при обработке CL-файла могут встретиться только в том случае, если содержащиеся в нем типы записей не известны постпроцессору.

Получение управляющей программы

Единицы измерения

Перемещения могут выводиться в УП в виде абсолютных значений или в приращениях. Могут использоваться как метрические единицы измерения, так и дюймовые. В любом случае эта информация должна быть задана.

Пример

<i>define codes</i>	<i>определить коды</i>
<i>metric units = G5 70</i>	<i>метрические единицы</i>
<i>imperial units = G5 71</i>	<i>дюймовые единицы</i>
<i>absolute = G6 90</i>	<i>абсолютные перемещения</i>
<i>incremental = G6 91</i>	<i>перемещения в приращениях</i>
<i>end define</i>	<i>конец определения</i>
<i>units = metric</i>	<i>устанавливаются метрические единицы</i>
<i>coordinates = absolute</i>	<i>абсолютные координаты</i>

Начало и конец УП

Символы и коды в начале и конце УП могут различаться для различных станков даже с одинаковыми системами ЧПУ. Некоторые системы ЧПУ требуют, чтобы в начале УП находился номер программы или ее имя. Их можно запрашивать во время выполнения постпроцессора или брать из команды APT PARTNO, используя переменные **PartId** (имя) и **ProgId** (номер). Специальные символы для начала и конца программы задаются в формате кадра начала и конца УП. Символы надо задавать в кавычках.

Ниже приводится список переменных, которые используются для этих целей.

TapeUnits	Используется для вывода кода используемых единиц измерения (обычно G70/G71 или G20/G21).
TapeCoords	Используется для вывода кода типа перемещений - абсолютных или в приращениях (обычно G20/G21).
PartID	Имя программы.
ProgID	Номер программы.
FromX, FromY, FromZ	Координаты X, Y и Z стартовой позиции.

Используются следующие ключевые слова:

N_	(Номер кадра)
ID_	(Номер программы)
G1 G2 G3	(Коды G)

Пример 1

Настройка заголовка программы .
Чтобы получить заголовок вида:

```
%
:0001
N10G91G28X0Y0Z0
N20G40G17G80G49
N30G0G90Z30.
```

Надо задать следующие настройки:

```
define block tape start
  "%"
  ID ProgID
  N ; G2 91 ; G6 28 ; X 0 ; Y 0 ; Z 0
  N ; compensation ; G6 49 ; xy plane ; G4
  N ; G5 90 ; G1 0 ; Z FromZ
end define
```

Пример 2

```
use progid = false
define block tape start           опред. кадр начала УП
  "%"
  Id_ ProgID
  N_ =P ; G1
  90
end define                       конец определения
define block tape end           определить кадр конца УП
  N_ =P ; M1 30
  "%"
end define                       конец определения
```

Стартовая позиция

Стартовая позиция инструмента появляется в УП либо когда задан кадр со стартовой позицией, либо когда в кадре смены инструмента используются переменные **FromX**, **FromY** и **FromZ**.

Пример

<pre>define block move from G1 92 ; X_ FromX ; Y_ FromY ; Z_ FromZ end define</pre>	<p><i>определить кадры перемещения в точку FROM конец определения</i></p>
---	---

Номера кадров

Номер кадра используется для идентификации кадров УП. Первый кадр УП может иметь в качестве номера 1, либо любое другое число. Если номер кадра выводить не требуется, его надо задать как "not permanent".

Пример

<pre>define word N_ adress letter = "N" field width = 4 permanent end define define keys blocknumber = N_ end define block start = 10 block increment = 10 maximum block number = 1000</pre>	<p><i>определить слово буква адреса ширина поля присутст. в каждом кадре конец определения определить ключи ключ номера кадра конец определения начальный номер кадра приращение номера максимальный номер</i></p>
--	--

Вывод сообщений

Сообщения (для которых в DUCT используется команда APT PPRINT) выводятся в УП с использованием логической переменной вывода. Необходимо определить символ начала и конца сообщения (обычно это круглые скобки).

Пример 1

Подавление вывода комментариев (PowerMILL вставляет достаточно много комментариев и, если не решен вопрос с DNC, их приходится подавлять).

```
message output = false
```

Пример 2

```
message output = true
define keys
  message start = MS_
  message end = EM_
end define
```

*разрешить вывод сообщений
определить ключи
начало сообщения
конец сообщения
конец определения*

Промежутки

Большинство систем ЧПУ позволяет вставлять в УП пробелы, хотя это нежелательно, т.к. увеличивает длину УП. Тем не менее УП становится более удобной для чтения. Количество пробелов между словами определяется с помощью определения формата вывода УП и определения формата вывода на печать. Если позиция фиксируется, может использоваться отрицательное значение, но в этом случае есть опасность писать поверх предыдущих слов.

Пример

```
define format (...)  
  tape position = 1  
  print position = -20  
end define
```

*определить формат
позиция вывода в УП
позиция вывода на печать
конец определения*

Это определение обеспечивает один пробел между словом и предыдущим словом в управляющей программе, а также вывод слова на печать, начиная с позиции 20 в строке.

Разбиение УП на несколько частей

УП можно разбить на несколько файлов или разделить на сегменты в одном файле с помощью специальных символов. Предварительно надо присвоить значение целой переменной, содержащей максимальную длину перфоленты или максимальную длину сегмента, а затем определить кадр, который будет выводиться в начале и конце каждой части. Расстояние, на которое отводится инструмент между обработкой отдельных частей УП, задается в целой переменной **withdrawal amount**. Значение переменной **maximum segment** определяется значением переменной **segment type** (0 означает длину перфоленты в футах, 1 означает длину отвода инструмента, 2 означает количество кадров).

Пример 1

Разрезка программ по заданному объему или по числу кадров. В первой строке задана разрезка на части по 11 КБ (на самом деле указана величина 90.0 – длина перфоленты в футах, но эта величина прямо пропорциональна размеру программы), во второй – по 950 кадров (но эта строка закрыта символом комментария). Надо использовать только один вариант разрезки. При разрезке в конце куска программы и в начале следующего куска вставляются необходимые кадры (подвод/отвод инструмента, включение/выключение шпинделя и т.п.). Имена файлов одинаковы, а расширения задаются так: у первого файла tap, далее t01, t02, t03 и т.д.

```
max tape length = 90.0
# max tape block = 950
```

Пример 2

<i>maximum tape length =</i>	<i>максимальная длина ленты</i>
<i>withdrawal amount =</i>	<i>длина отвода</i>
<i>define block tape split start</i>	<i>опред. кадр начала</i>
<i>разбиения</i>	
.....	
<i>end define</i>	<i>конец определения</i>
<i>define block tape split move</i>	<i>опред. кадр перемещения</i>
.....	
<i>end define</i>	<i>конец определения</i>
<i>define block tape split end</i>	<i>опред. кадр конца разбиения</i>
.....	
<i>end define</i>	<i>конец определения</i>
<i>или</i>	
<i>segment type =</i>	<i>тип сегментирования</i>
<i>maximum segment =</i>	<i>макс. длина сегмента</i>
<i>define block tape segment</i>	<i>опред. кадр разбиения</i>
.....	
<i>end define</i>	<i>конец определения</i>

Шпиндель и охлаждение

Для вывода в УП кадров управления шпинделем и охлаждением необходимо определить соответствующие ключи и коды.

Пример

<i>spindle output = (.....)</i>	<i>вывод информ. о шпинделе</i>
<i>coolant output = (.....)</i>	<i>вывод информ. об охлажд.</i>
<i>define keys</i>	<i>определить ключи</i>
<i>spindle</i>	<i>шпиндель</i>
<i>end define</i>	<i>конец определения</i>
<i>define codes</i>	<i>определить коды</i>
<i>coolant on = M1 8</i>	<i>включить охлаждение</i>
<i>coolant off = M1 9</i>	<i>выключить охлаждение</i>

<i>spindle on = M1 3</i>	<i>включить шпиндель</i>
<i>spindle on cc = M1 4</i>	<i>включить шпиндель против часовой стрелки</i>
<i>spindle off = M1 5</i>	<i>выключить шпиндель</i>
<i>spindle and coolant on = M1 13</i>	<i>включить шпиндель и охлаждение</i>
<i>end define</i>	<i>конец определения</i>

Смена инструмента

Очень часто изменения в файле опций касаются кадра смены инструмента. Обычно при смене инструмента требуется указать код смены инструмента и код номера инструмента. Иногда бывает необходимо задать коды смещений на длину и на радиус инструмента. Для формирования кадра смены инструмента используются следующие переменные:

ToolNumber	Номер инструмента из команды APT LOADTL.
ToolLenOff	Номер корректора на длину.
ToolRadOff	Номер корректора на радиус.
NextTool	Номер следующего используемого инструмента. (для станков с предварительным выбором).
ToolLength	Длина инструмента.
ToolFeed	Скорость подачи текущего инструмента.
ToolSpeed	Скорость вращения шпинделя текущего инструмента.

Используются также слова

T_	(Номер инструмента)
H_	(Номер корректора на длину)
D_	(Номер корректора на радиус)

Кадр для первой установки инструмента можно определить отдельно от кадра для последующих смен инструмента. При последующих сменах инструмента можно добавить выключение шпинделя, перемещение в безопасную позицию, автоматическую коррекцию на длину.

Пример

<i>define block tool change first</i>	<i>определение кадра первой установки инструмента</i>
<i> N_ =P ; T_ =S ; M1 6 G6 90 ;</i>	
<i>end define</i>	<i>конец определения</i>
<i>define block tool change</i>	<i>определение</i>

	<i>кадра смены инструмента</i>
<i>N_ =P ; H_ 0 ; M1</i>	
<i>9 N_ =P ; T_ =S ;</i>	
<i>M1 6</i>	
<i>end define</i>	<i>конец определения</i>

Линейные перемещения

Необходимо определить ключи для координат x, y и z, а также задать вещественную переменную для максимальной скорости подачи и кадр линейного перемещения. Кадр линейного перемещения должен включать все слова, которые могут в нем появиться (например, функции M и скорость вращения шпинделя). Функцию G линейного перемещения (обычно G01) можно определить здесь, либо в секции определения кодов. Могут понадобиться и другие функции G (для компенсации инструмента и т.д.).

Пример

<i>define keys</i>	<i>определить ключи</i>
<i> x coord = X_</i>	<i>координата X</i>
<i> y coord = Y_</i>	<i>координата Y</i>
<i> z coord = Z_</i>	<i>координата Z</i>
<i>end define</i>	<i>конец определения</i>
<i>maximum feedrate = 9999.</i>	<i>макс. скорость подачи</i>
<i>define block move linear</i>	<i>определение кадра линейного перемещения</i>
<i> N_ =P ; G1 1 ; X_ =S ; Y_ =S ; Z_ =S ;</i>	
<i> G2 =S ; M1 =S ; S_ =S</i>	
<i>end define</i>	<i>конец определения</i>

Круговая интерполяция

Необходимо задать массив условий вывода кадров круговой интерполяции и определить кадр для круговой интерполяции. Коды функций G при движении по и против часовой стрелки задаются в секции определения кодов. Если круговая интерполяция выполняется в плоскости, отличной от плоскости XY, можно задать коды соответствующей плоскости.

Пример

<i>define codes</i>	<i>определить коды</i>
<i> circle cw = G1 2</i>	<i>круговая интерполяция по</i>

<code>circle ccw = G1 3</code>	часовой стрелке круговая интерполяция против часовой стрелки
<code>xy plane = G3 17</code>	плоскость интерполяции xy
<code>zy plane = G3 18</code>	плоскость интерполяции zy
<code>xz plane = G3 19</code>	плоскость интерполяции
<code>xz end define</code>	конец определения
<code>circle output = (1 1 1 1)</code>	массив условий вывода круговой интерполяции
<code>define block move circle</code>	определение кадра круговой интерполяции
<code>N_ =P ; G1 =S ; G2 =S ; G3 =S ; X_ =S ; Y_ =S ; Z_ =S ; I_ =S ; J_ =S ; Z_ =S ; M1 =S ; S_ =S</code>	
<code>end define</code>	конец определения

Быстрые перемещения

Необходимо задать кадр быстрого перемещения. Если требуется, в кадре можно задать подачу или код подачи. Вещественная переменная, содержащая скорость быстрого перемещения, используется только для вычисления времени, затраченного на перемещение.

Пример

<code>define block move rapid</code>	определение кадра быстрого перемещения
<code>N_ =P ; G1 =S ; X_ =S ; Y_ =S ; Z_ =S ;</code>	
<code>end define</code>	конец определения

Циклы сверления

Если система ЧПУ допускает использование циклов сверления, необходимо присвоить целой переменной **cycle output** значение 1 (если переменная будет иметь значение 0, в УП будет выводиться каждое движение, выполняемое при сверлении). В секции определения кодов надо определить коды для каждого цикла сверления. Необходимо также определить ключи для глубины, величины шага, времени задержки и высоты отвода и кадры, содержащие циклы сверления. Обычно общие циклы сверления определяются отдельно от циклов нарезания резьбы из-за значительных различий.

Пример

<i>define keys</i>	определить ключи
<i>clearplane = R</i>	высота отвода
<i>drill peck depth = Q_</i>	глубина одного шага
<i>drill hole depth = Z2</i>	глубина отверстия
<i>cycle dwell = P_</i>	время задержки
<i>end define</i>	конец определения
<i>define codes</i>	определить коды
<i>end of drill = G4 80</i>	конец цикла
<i>drill = G4 81</i>	сверление
<i>break chip =</i>	дробление стружки
<i>deep drill</i>	глубокое сверление
<i>tap</i>	нарезание резьбы
<i>bore 1</i>	расточка 1
<i>bore 2</i>	расточка 2
<i>bore 3</i>	расточка 3
<i>bore 4</i>	расточка 4
<i>bore 5</i>	расточка 5
<i>end define</i>	конец определения
<i>define block cycle start</i>	опред. кадра начала цикла.
<i>N_ =P ; G4 =S ; Z2</i>	
<i>=S ; R_ =S ; Q_ =S ;</i>	
<i>P_ =S ;</i>	
<i>end define</i>	конец определения
<i>define block move cycle</i>	опред. кадра перемещения
<i>N_ =P ; G4 79 ; X_ =S ; Y_ =S ; Z_ =S</i>	в цикле
<i>end define</i>	конец определения
<i>define block cycle end</i>	опред. кадра конца цикла.
<i>N_ =P ; G4 80 ;</i>	
<i>end define</i>	конец определения
<i>define block move tap</i>	опред. кадра начала цикла.
<i>N_ =P ; G4 =S ; X_ =S ; Y_ =S ;</i>	нарезания резьбы
<i>Z_ =S ; Z2 =S ; R_ =S ; Z2 =S</i>	
<i>end define</i>	конец определения

Перемещение G0H0ME

Перемещение по команде APT G0H0ME будет выведено в УП в том случае, если установлена логическая переменная **go home output** и определены кадры **go home xy move** и **go home z move**.

Пример

<i>go home output = true</i>	разрешение вывода G0H0ME
<i>define block go home xy move</i>	определение кадра с
<i>N_ =P ; G1 0 ; X_ =S ; Y_ =S</i>	перемещ. G0H0ME по xy
<i>end define</i>	конец определения
<i>define block go home z move</i>	определение кадра с
<i>N_ =P ; G1 0 ; Z_ =S</i>	перемещ. G0H0ME по z
<i>end define</i>	конец определения

Приложение А

Список ключей

3rd rotation axis	третья ось вращения
aux function	вспомогательная функция
azimuth axis	ось азимута
blocknumber	номер кадра
clearplane	высота отвода для циклов сверления
drill hole depth	глубина отверстия при сверлении
drill peck depth	глубина одного шага при сверлении
dwel	время задержки
elevation axis	ось элевации
feedrate	скорость подачи
feedrate per revolution	скорость подачи на оборот
key i	ключ i-составляющей для круговой
key j	ключ j-составляющей интерпо-
key k	ключ k-составляющей ляции
message end	конец сообщения для вывода
message start	начало сообщения PPRINT
opskip	опция пропуска кадров
preparatory function	подготовительная функция
program id	идентификатор программы
radius	радиус
spindle	шпиндель
tool length	длина инструмента
tool number	номер инструмента
tool radius	радиус инструмента
x coordinate	координата x
y coordinate	координата y
z coordinate	координата z

Приложение Б

Список кодов функций

Постпроцессор распознает следующие коды функций:

absolute data	абсолютные координаты
bore1	расточка отверстия 1
bore2	расточка отверстия 2
bore3	расточка отверстия 3
bore4	расточка отверстия 4
bore5	расточка отверстия 5
break chip	дробление стружки

change tool	смена инструмента
circle ccw	круговая интерполяция против часовой стрелки
circle cw	круговая интерполяция по часовой стрелке
clamp off	включить зажим
clamp on	выключить зажим
compensation off	выключить компенсацию на радиус инструмента
compensation on left	включить компенсацию слева
compensation on right	включить компенсацию справа
constant surface speed	постоянная линейная скорость (для токарной
обраб.)	
coolant off	выключить охлаждение
coolant on	включить охлаждение
coolant on flood	включить охлаждение жидкостью
coolant on mist	включить капельное охлаждение
cycle retract	отвод инструмента внутри цикла
deep drill	глубокое сверление
drill	сверление
dwel	задержка
end of drill	конец сверления
end of prog	конец программы
end of tape	конец УП
feedrate per minute	скорость подачи в минуту
feedrate per revolution	скорость подачи на оборот
from	стартовая позиция обработки
gear range 1	диапазон скорости вращения шпинделя 1
gear range 2	диапазон скорости вращения шпинделя 2
gear range 3	диапазон скорости вращения шпинделя 3
imperial data	дюймовые единицы измерения
incremental data	координаты в приращениях
linear	линейное перемещение
metric data	метрические единицы измерения
opt stop	необязательный останов
rapid	ускоренная подача
spin coolant off	выключение шпинделя и охлаждения
spin coolant on ccw	включение шпинделя против часовой стрелки и
	включение охлаждения
spin coolant on cw	включение шпинделя по часовой стрелке и
	включение охлаждения
spindle off	выключение шпинделя
spindle on cc	включение шпинделя против часовой стрелки
spindle on cw	включение шпинделя по часовой стрелке
spindle rpm	скорость вращения шпинделя в оборотах в минуту
stop	останов
tap	нарезание резьбы
tool length offset	коррекция на длину инструмента
xy plane	плоскость круговой интерполяции xy
xz plane	плоскость круговой интерполяции xz
zy plane	плоскость круговой интерполяции zy