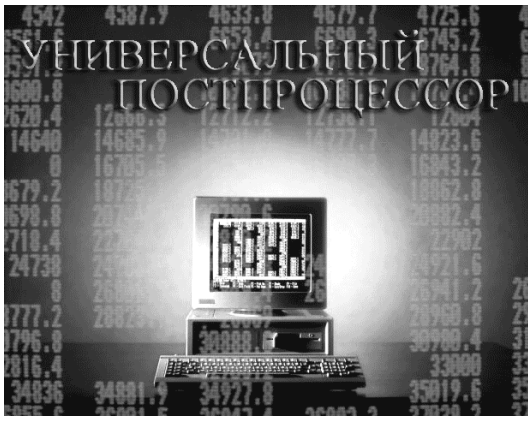


Глава 4

Параметры постпроцессора и исходной программы



Параметры постпроцессора и исходной программы

Параметры константы, задаваемые пользователем в разделе инициализации

Параметры, задаваемые пользователем, такие как дискретность стойки УЧПУ, максимально допустимый радиус круговой интерполяции и некоторые другие, представляют собой величины, определяемые в разделе инициализации, и используются интерпретатором для различного рода вычислений при анализе исходной программы. Данные параметры являются константами и действуют как в главной программе, так и подпрограммах. При описании данных параметров должен использоваться оператор **CONST.**

RDISCR

Тип

вещественный

Умолчание

0.001

Описание

Дискретность представления величин, характеризующих протяженность, таких как координаты точек, радиусы дуг и т.д. Определяет точность, с которой подобные параметры будут вычисляться интерпретатором, и единицу измерения, в которой они будут представлены при анализе исходной программы. Например, если в исходной программе есть команда перемещения в точку с координатами (10.5, 20.8, 30), параметры x, y, z этой команды будут иметь значения соответственно 10500, -20800, 30000 для RDISCR = 0.001 и 1050, -2080, 3000 для RDISCR = 0.01. Если в исходной программе линейные величины заданы с точностью большей, чем значения RDISCR, соответствующие параметры будут округлены. Параметр RDISCR определяет точность представления параметров x,y,z, dx,dy,dz, i,j,k, di,dj,dk, r, x0,y0,z0, shift_x0,shift_y0,shift_z0, impt_x,impt_y,impt_z.

Пример

```
CONST RDISCR = 0.01  'присвоение параметру RDISCR значения
                     'дискретности по перемещениям равной 0.01 мм
```

RMAX

Тип

вещественный

Умолчание

9999999.

Описание

Выраженный в единицах дискретности (дискретах) максимально допустимый радиус дуги при круговой интерполяции. Если в процессе чтения исходной программы интерпретатор обнаружит перемещение по дуге большего радиуса, она будет интерполирована серией линейных перемещений (хорд этой дуги) с точностью прогиба, равной заданной параметром ARCDISCR дискретности (если он не задан RDISCR) . Таким образом, вместо одной команды с кодом MOVE# и параметром primitive#, равным ARC_CLW# или ARC_CCLW#, интерпретатор будет анализировать серию команд с кодом MOVE# и параметром primitive#, равным LINE. (см описания соответствующих параметров).

Пример

```
CONST RMAX = 999999.0 ' Если дуга, заданная в исходной
программе
                      ' имеет радиус, больший 999999.0 единиц
                      ' дискретности, то она будет заменена
                      ' последовательностью линейных перемещений
```

ARCDISCR

Тип

вещественный

Умолчание

Совпадает с RDISCR

Описание

Определяет точность прогиба дуги относительно ее хорды при разбиении дуги на отрезки.

Пример

```
CONST ARCDISCR = 0.01 ' присвоение параметру ARCDISCR значения
0.01
```

---

emul_rapid#

Тип

логический

Умолчание

FALSE#

Описание

Если заданное значение параметра равно TRUE#, и в процессе чтения исходной программы интерпретатор обнаружит команду позиционирования, она будет заменена линейным перемещением в ту же точку с подачей, устанавливаемой параметром RFEED. Таким образом, вместо одной команды с кодом MOVE# и параметром primitive#, равным

LOCATE#, интерпретатор будет анализировать команду с кодом MOVE#, параметром primitive#, равным LINE#, и параметром f, равным значению параметра RFEED. (см. описания соответствующих параметров ниже). Параметр следует задавать для УЧПУ, не поддерживающих режим позиционирования (быстрых перемещений), например, КУРС-332.

RFEED

Тип

вещественный

Умолчание

не определено

Описание

Подача позиционирования. Если заданное значение параметра emul_rapid# равно TRUE#, команды позиционирования будут эмулироваться командами линейных перемещений с данной подачей.

Пример

```
CONST emul_rapid# = TRUE# ' Установка параметра emul_rapid# в
CONST TRUE#
RFEED = 1200    ' приводит к эмуляции команд позиционирования
                ' линейными перемещениями с подачей,
                ' определяемой величиной RFEED
```

---

emul_q#

Тип

логический

Умолчание

FALSE#

Описание

Если заданное значение параметра равно TRUE#, и в процессе чтения исходной программы интерпретатор обнаружит перемещение по дуге, она будет при необходимости разбита на серию дуг того же радиуса, каждая из которых будет лежать внутри одного квадранта. Таким образом, вместо одной команды с кодом MOVE# и параметром primitive#, равным ARC_CLW# или ARC_CCLW#, интерпретатор будет анализировать серию команд с кодом MOVE# и параметром primitive#, равным ARC_CLW# или ARC_CCLW (см. описания соответствующих параметров ниже). Параметр следует задавать для УЧПУ, допускающих круговые перемещения только по дугам, лежащим внутри одного квадранта, например, КУРС-332.

Пример

```
CONST emul_q# = TRUE# ' Установка параметра emul_q# в TRUE#
                ' приводит к разбиению на квадранты дуг,
                ' заданных в исходной программе
```

XDISCR

Тип

вещественный

Умолчание

0.001

Описание

Дискретность представления дополнительных 4-й, 5-й и 6-й координат **u, v, w** в командах перемещения. Определяет точность, с которой подобные параметры будут вычисляться интерпретатором, и единицу измерения, в которой они будут представлены при анализе исходной программы. Например, если в исходной программе есть команда перемещения в точку с 4-й, 5-й и 6-й координатами (15.5, 25.8, 35), параметры u, v, w этой команды будут иметь значения соответственно 15500, -25800, 35000 для XDISCR = 0.001 и 1550, -2580, 3500 для XDISCR = 0.01. Если в исходной программе дополнительные координаты заданы с точностью большей, чем значения XDISCR, соответствующие параметры будут округлены. Параметр XDISCR определяет точность представления параметртов u, v, w, du, dv, dw, u0, v0, w0, shift_u0, shift_v0, shift_w0, impt_u, impt_v, impt_w.

---

d5_mode#

Тип

логический

Умолчание

FALSE#

Описание

Если заданное значение параметра равно TRUE#, и в процессе чтения исходной программы интерпретатор обнаружит перемещение только по дополнительным координатам **u,v,w**, то эта команда с кодом MOVE# будет обработана интерпретатором постпроцессоров. В противном случае, она будет проигнорирована.

Если заданное значение параметра равно TRUE#, и в процессе чтения исходной программы интерпретатор обнаружит два или более линейных перемещений вдоль одной прямой в одном направлении, то каждое из этих перемещений будет обработано интерпретатором постпроцессоров раздельно при условии, что изменяются дополнительные координаты u,v,w. В противном случае эти перемещения будут объединены в единую команду с кодом MOVE#

---

d2_5_mode#

Тип

логический

Умолчание

FALSE#

Описание

Если заданное значение параметра равно TRUE#, то в процессе чтения исходной программы перемещения по трем координатам будут разбиты на два перемещения.

При опускании (dz < 0), сначала будет выполнено движение в плоскости XY, затем опускание по Z.

При подъеме (dz > 0), сначала будет выполнен подъем по Z, затем движение в плоскости XY.

SUB_BEGIN#

Тип

логический

Умолчание

FALSE#

Описание

Если заданное значение параметра равно TRUE#, то все подпрограммы управляющей программы автоматически будут располагаться ее начале.

Если заданное значение параметра равно FALSE#, то все подпрограммы автоматически будут располагаться в конце программы.

AUTO_LOAD#

Тип

логический

Умолчание

FALSE#

Описание

Параметр предназначен для упрощения составления постпроцессоров для станков с автоматической сменой инструмента в спецточке станка

Если заданное значение параметра равно TRUE#, то считается, что при смене инструмента станок уходит в спецточку. Происходит изменение текущих координат x,y,z.

Параметры анализируемой (предшествующей, последующей) команды исходной программы

Параметры команд представляют собой значения, вычисляемые интерпретатором при анализе исходной программы. Эти значения могут использоваться в операторах главного раздела описания при дополнительном анализе команды и выводе информации в управляющую программу.

Код команды

---

command# [command_#, _command#]

Тип

перечислимый: DO_NOTHING#, HEADER#, LOAD_TOOL#, CALL#, MOVE#, STOP#,OPSTOP#, PROGRAM#, END#,ENDALL#, PROCEDURE#, RETURN#, SET_ORIGIN#, DO_CYCLE#, CYCLE#, DELAY#, ROTATE #,ORIGNO#,INSERT#

Умолчание

DO_NOTHING#

Описание

Идентифицирует анализируемую [предшествующую, последующую] команду исходной программы. Т.е. представляет собой код анализируемой [предшествующей, последующей] команды. Значения параметра command# - константы перечислимого типа имеют следующий смысл:

DO_NOTHING#	пустая команда
HEADER#	Заголовок, является первой командой при анализе файла с исходной программой
LOAD_TOOL#	Команда загрузки инструмента в шпиндель
CALL#	вызов подпрограммы (номер вызываемой подпрограммы определяется параметром unit_call#
MOVE#	Команда перемещения (вид перемещения определяется параметром primitive#)
STOP#	Команда останова
OPSTOP#	Команда условного останова
PROGRAM#	Команда начала главной программы
END#	Команда окончания главной программы.
ENDALL#	Команда окончания файла исходной программы
PROCEDURE#	Команда начала подпрограммы (номер анализируемой подпрограммы определяется параметром proc_no#
RETURN#	Команда возврата из подпрограммы
ORIGNO#	Команда задания номера станочной системы координат
INSERT#	Команда вставки строки из исходного APT файла
DO_CYCLE#	Команда "выполнить цикл"
CYCLE#	Команда "задать цикл"
DELAY#	Команда останова на заданное время
ROTATE#	Команда поворота стола
SET_ORIGIN#	Команда определения первой точки движения

Пример

```
' Обработка кода анализируемой команды исходной программы
if command# = HEADER# then
' операторы, выполняемые для команды "заголовок", первой
' команды при анализе файла с исходной-программой
elseif command# = LOAD_TOOL# then
' операторы, выполняемые для команды
' загрузки инструмента в шпиндель
elseif command# = MOVE# then
' операторы, выполняемые для команды
' перемещения
elseif command# = STOP# then
' операторы, выполняемые для команды
' останова
elseif command# = PROGRAM# then
' операторы, выполняемые для команды
' начала главной программы
elseif command# = END# then
' операторы, выполняемые для команды
' окончания главной программы и
' файла исходной программы
elseif command# = PROCEDURE# then
' операторы, выполняемые для команды
' начала подпрограммы
elseif command# = RETURN# then
' операторы, выполняемые для команды
' возврата из подпрограммы
elseif command# = CALL# then
' операторы, выполняемые для команды
' "вызов подпрограммы"
elseif command# = SET_ORIGIN# then
' операторы, выполняемые для команды
' установки новых значений
' координат текущей точки исходной
' программы
endif
' окончание обработки анализируемой команды исходной программы
```

Параметры инструмента

Общее замечание , касающееся всех параметров инструмента: При анализе команды смены инструмента значение параметра становится равным номеру инструмента, задаваемого этой командой. Значение сохраняется неизменным до тех пор, пока не будет переопределено новой командой смены инструмента.

---

cutter_no# [cutter_no_#, _cutter_no#]

Тип

целый

Умолчание

0

Описание

Номер заданного в исходной программе инструмента для анализируемой [предшествующей, последующей] команды исходной программы.

Пример

```
if command# = LOAD_TOOL# then      'Вывод информации,
                                   'связанной со сменой
инструмента.
write id1# using "N%d", n#         ' Вывод номера текущего
кадра
n# = n# + 1                        ' Увеличиваем счетчик
кадров
write id1# using "T%.3dM06", cutter_no# ' Вывод номера
                                   ' инструмента и M06
write id1# using "%s", NL$         ' Завершение кадра
```

---

cutter_diameter [cutter_diameter_, _cutter_diameter]

Тип

вещественный

Умолчание

0

Описание

Диаметр заданного в исходной программе инструмента для анализируемой [предшествующей, последующей] команды исходной программы.

---

cutter_radius [cutter_radius_, _cutter_radius]

Тип

вещественный

Умолчание

0

Описание

Радиус заданного в исходной программе инструмента для анализируемой [предшествующей, последующей] команды исходной программы.

---

cutter_rf [cutter_rf_, _cutter_rf]

Тип

вещественный

Умолчание

0

Описание

Радиус скругления заданного в исходной программе инструмента для

анализируемой [предшествующей, последующей] команды исходной программы.

---

cutter_length [cutter_length_, _cutter_length]

Тип

вещественный

Умолчание

0

Описание

Длина заданного в исходной программе инструмента.

---

cutter_conus [cutter_conus_, _cutter_conus]

Тип

вещественный

Умолчание

0

Описание

Конусность заданного в исходной программе инструмента.

---

cutter_dt [cutter_dt_, _cutter_dt]

Тип

вещественный

Умолчание

0

Описание

Диаметр торца заданного в исходной программе инструмента.

Режим и обороты шпинделя

---

sm# [sm_#, _sm#]

Тип

перечислимый: SPIN_OFF#, SPIN_CLW#, SPIN_CCLW#

Умолчание

SPIN_OFF#

Описание

Режим вращения шпинделя, при котором в соответствии с исходной программой должна выполняться анализируемая [предшествующая,последующая] команда исходной программы. Принимает значения перечислимого типа, имеющие следующий смысл:

SPIN_CLW#, SPIN_CCLW# - в исходной программе на текущий момент задан режим вращения шпинделя по или против часовой стрелки, соответственно.

SPIN_OFF# - в исходной программе на текущий момент вращение шпинделя выключено.

---

s [s_, _s]

Тип

вещественный

Умолчание

0

Описание

Частота вращения шпинделя, при которой в соответствии с исходной программой должна выполняться анализируемая [предшествующая, последующая] команда исходной программы. Значение определено, только если параметр sm# для данной команды имеет значения SPIN_CLW# или SPIN_CCLW#.

Пример

```
' Если режим вращения шпинделя изменился, т.е.
' значение параметра sm# отличается от значения
' аналогичного параметра предшествующей команды sm_# :
if sm# <> sm_# then
write id1# using "N%d", n# ' Вывод номера текущего кадра
n# = n# + NSTEP# ' Увеличиваем счетчик кадров
' Вывод функции направления вращения шпинделя
' в зависимости от значения параметра sm# :
if sm# = SPIN_CLW# then
' Вывод оборотов шпинделя по адресу S
' и функции M03 :
write id1# using "S%.15gM03", s
elseif sm# = SPIN_CCLW# then
' Вывод оборотов шпинделя по адресу S
' и функции M04 :
write id1# using "S%.15gM04", s
elseif sm# = SPIN_OFF# then
' Вывод функции M05 :
write id1# using "M05"
endif
write id1# using "%s", NL$ ' Завершение кадра - символ LF
endif
```

Подача

f [f_, _f]

Тип

вещественный

Умолчание

0

Описание

Величина подачи, заданная в исходной программе на момент выполнения анализируемой [предшествующей, последующей] команды исходной программы.

Пример

```
if (primitive# <> LOCATE#) AND (f <> f_) then
' Вывод подачи, если перемещение не является движением с GO O
' и на данном перемещении задано новое значение подачи.
' Обратите внимание на расстановку скобок в условии.
write id1# using "F%.15g", f
endif
```

Режим охлаждения

cool# [cool_#, _cool#]

Тип

перечислимый: OFF#, ON#

Умолчание

OFF#

Описание

Режим охлаждения шпинделя, при котором в соответствии с исходной программой должна выполняться анализируемая [предшествующая, последующая] команда исходной программы. Принимает значения перечислимого типа, имеющие следующий смысл:

ON# - в исходной программе на текущий момент охлаждение включено

OFF# - в исходной программе на текущий момент охлаждение выключено.

Пример

```
' Если режим охлаждения изменился, т.е.
' значение параметра cool# отличается от значения
' аналогичного параметра предшествующей команды coo1_:
if cool#<>cool_# then
write id1# using "N%d", n#       ' Вывод номера текущего кадра
n# = n# + NSTEP#                 ' Увеличиваем счетчик кадров
```

```
' Вывод функции направления вращения шпинделя
' в зависимости от значения параметра cool#:
if cool# = ON# then
write id1# using "M08"
elseif cool# = OFF# then
write id1# using "M09"
endif
write id1# using "%s", NL$      ' Завершение кадра - символ LF
endif
```

Вид движения

---

primitive# [primitive_#, _primitive#]

Тип

перечислимый: GO_OFF#, LOCATE#, LINE#, ARC_CLW#, ARC_CCLW#

Умолчание

GO_OFF#

Описание

Значение параметра соответствует виду перемещения, если анализируемая [предшествующая, последующая] команда исходной программы или подпрограммы есть команда перемещения. Принимает значения перечислимого типа, имеющие следующий смысл:

GO_OFF# - режим перемещения не был определен в исходной программе или подпрограмме к моменту анализа данной команды (используется в специальных случаях, для анализа команд не связанных с движением)

LOCATE# - позиционирование

LINE# - движение по прямой

ARC_CLW#, ARC_CCLW# - движение по дуге окружности по и против часовой стрелки соответственно.

Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой перемещения.

Пример

```
' если на анализируемом перемещении
' изменился режим движения
' (т.е. параметр primitive# анализируемой команды
' не равен аналогичному параметру предшествующей
' команды primitive_#) :
if primitive# <> primitive_# then
' Вывод соответствующей G-функции линейной
' или круговой интерполяции :
if primitive# = LOCATE# then
write id1# using "GOO"
elseif primitive# = LINE# then
write id1# using "G01"
```

```
elseif primitive# = ARC_CLW# then
write id1# using "G02"
elseif primitive# = ARC_CCLW# then
write id1# using "G03"
endif
```

Плоскость круговой интерполяции

plane# [plane_#, _plane#]

Тип

перечислимый: PLANE_XY#, PLANE_YZ#, PLANE_ZX#

Умолчание

PLANE_XY#

Описание

Значение параметра соответствует заданной плоскости круговой интерполяции для анализируемой [предшествующей, последующей] команды исходной программы. Принимает значения перечислимого типа, имеющие следующий смысл:

PLANE_XY# - плоскость интерполяции XY

PLANE_YZ# - плоскость интерполяции YZ

PLANE_ZX# - плоскость интерполяции ZX .

Плоскость перемещения

m_plane# [m_plane_#, _m_plane#]

Тип

перечислимый: PLANE_XY#, PLANE_YZ#, PLANE_ZX#

Умолчание

PLANE_XY#

Описание

Значение параметра соответствует заданной плоскости перемещения для анализируемой [предшествующей, последующей] команды исходной программы. Принимает значения перечислимого типа, имеющие следующий смысл:

PLANE_XY# - плоскость перемещения XY

PLANE_YZ# - плоскость перемещения YZ

PLANE_ZX# - плоскость перемещения ZX .

Параметры перемещения рабочего органа

---

x, y, z [x_, y_, z_, _x, _y, _z]

Тип

вещественный

Умолчание

параметры x0, y0, z0 (координаты начальной точки)

Описание

Для команд движения параметры равны выраженным в **дискретах** координатам x,y,z конечной точки анализируемого [предшествующего, последующего] движения. В остальных случаях - координатам текущей точки анализируемой [предшествующей, последующей] команды исходной программы. Дискретность этих параметров задается величиной RDISCR.

---

dx, dy, dz [dx_ ,dy_ ,dz_, _dx, _dy, _dz]

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Для команд движения параметры равны выраженным в **дискретах** разностям координат конечной и начальной точки анализируемого [предшествующего, последующего] движения. Значения параметров сохраняются неизменными до тех пор, пока не будут переопределены новой командой движения. Дискретность этих параметров задается величиной RDISCR.

Пример

```
' Запись по адресу X, если приращение этой координаты,
' dx, не равно 0:
if dx<>0.0 then
write id1# using "X%+.f", x * RDISCR
endif
' Запись по адресу Y, если приращение этой координаты,
' dy, не равно 0:
if dy<>0.0 then
write id1# using "Y%+.f", y * RDISCR
endif
' Запись по адресу Z, если приращение этой координаты,
' dz, не равно 0:
if dz<>0.0 then
```

```
 write id1# using "Z%+.f", z * RDISCR
endif
```

---

i, j, k [i_, j_, k_, _i, _j, _k]

Тип

вещественный

Умолчание

не определено

Описание

Выраженные в дискретах координаты центра дуги, если анализируемая [предшествующая, последующая] команда есть команда движения по окружности. Значения параметров сохраняются неизменными до тех пор, пока не будут переопределены новой командой движения по дуге. Дискретность этих параметров задается величиной RDISCR.

Пример

```
' Для круговой интерполяции вывод по адресам I, J, K:
if (primitive# = ARC_CLW#) OR (primitive# = ARC_CCLW#) then
 ' Если дуга лежит в плоскости XY выводятся I и J :
 if plane# = PLANE_XY# then
  write id1# using "I%.15g", i*RDISCR
  write id1# using "J%.15g", j*RDISCR
 ' Иначе, если дуга лежит в плоскости YZ, выводятся J и K:
 elseif plane# = PLANE_YZ# then
  write id1# using "J%.15g", j*RDISCR
  write id1# using "K%.15g", k*RDISCR
 ' Иначе, если дуга лежит в плоскости ZX, выводятся I и K:
 elseif plane# = PLANE_ZX# then
  write id1# using "I%.15g", i*RDISCR
  write id1# using "K%.15g", k*RDISCR
 endif
endif
```

---

di, dj, dk [di_, dj_, dk_, _di, _dj, _dk]

Тип

вещественный

Умолчание

не определено

Описание

Выраженные в дискретах разности координат центра дуги и начальной точки дуги, если анализируемая [предшествующая, последующая] команда есть команда движения по окружности. Значения параметров сохраняются неизменными до тех пор, пока не будут переопределены новой командой движения по дуге. Дискретность этих параметров задается величиной RDISCR.

Пример

```
' Для круговой интерполяции вывод по адресам I, J, K :
if (primitive# = ARC_CLW#) OR (primitive# = ARC_CCLW#) then
' Если дуга лежит в плоскости XY выводятся di  и dj:
if plane# = PLANE_XY# then
write id1# using "I%.15g", di*RDISCR
write id1# using "J%.15g", dj*RDISCR
' Иначе, если дуга лежит в плоскости YZ, выводятся dj и dk:
elseif plane# = PLANE_YZ# then
write id1# using "J%.15g", dj*RDISCR
write id1# using "K%.15g", dk*RDISCR
' Иначе, если дуга лежит в плоскости ZX, выводятся di и dk:
elseif plane# = PLANE_ZX# then
write id1# using "I%.15g", di*RDISCR
write id1# using "K%.15g", dk*RDISCR
endif
endif
```

---

r [r_, _r]

Тип

вещественный

Умолчание

не определено

Описание

Выраженный в дискретах радиус окружности, если анализируемая [предшествующая, последующая] команда есть команда движения по окружности. Значение параметра сохраняется неизменным до тех пор, пока не будут переопределены новой командой движения по дуге. Дискретность этого параметра задается величиной RDISCR.

Пример

```
' Для круговой интерполяции выводится радиус дуги по адресу
R:
if (primitive# = ARC_CLW#) OR (primitive# = ARC_CCLW#) then
write id1# using " R%.f", r*RDISCR
endif
```

---

impt_x, impt_y, impt_z [impt_x_, impt_y_, impt_z_, _impt_x, _impt_y, _impt_z]

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Выраженные в дискретах суммы импульсов (величины абсолютного физического перемещения) по координатам x,y,z текущей точки анализируемой [предшествующей, последующей] команды относительно

самой первой точки программы или подпрограммы. Дискретность этих параметров задается величиной RDISCR.

---

len [len_, _len]

Тип

вещественный

Умолчание

0

Описание

Для команд движения параметр равен длине траектории движения от начальной точки анализируемого [предшествующего, последующего] движения до конечной точки. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой движения.

---

al_beg [al_beg_, _al_beg]

Тип

вещественный

Умолчание

0

Описание

Для команд кругового движения параметр равен углу положения начальной точки анализируемого [предшествующего, последующего] движения. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой кругового движения.

---

al_end [al_end_, _al_end]

Тип

вещественный

Умолчание

0

Описание

Для команд кругового движения параметр равен углу положения конечной точки анализируемого [предшествующего, последующего] движения. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой кругового движения.

---

al_delt [al_delt_, _al_delt]

Тип

вещественный

Умолчание

0

Описание

Для команд кругового движения параметр равен углу поворота из начальной точки анализируемого [предшествующего, последующего] движения в конечную точку. Угол считается положительным, если траектория поворачивается против часовой стрелки, и отрицательным в противном случае. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой кругового движения.

Касательные вектора перемещений

---

tx0, ty0, tz0 [tx0_, ty0_, tz0_, _tx0, _ty0, _tz0]

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Для команд движения параметры равны выраженным в **дискретах** проекциям на координатные оси вектора, касательного траектории движения в начальной точке анализируемого [предшествующего, последующего] движения. Длина этого вектора принимается равной длине перемещения для линейных перемещений (в этом случае tx0=dx, ty0=dy, tz0=dz,) и радиусу дуги для перемещений по окружности. Направление касательного вектора совпадает с направлением движения в начальной точке перемещения. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой движения.

Пример

```
' Вывод по адресам I и J соответствующих проекций вектора,
' направление которого совпадает с касательной к
' траектории движения в начале следующего кадра (перемещения)
' а длина равна длине анализируемого перемещения
' (Hypot (dx, dy))
if _tx0 <> 0.0 then
write id1# using
"I%+f",Round(_tx0*Hypot(dx,dy)/Hypot(_tx0,ty0))
endif
if _ty0 <> 0.0 then
write id1# using
"J%+f",Round(_ty0*Hypot(dx,dy)/Hypot(_tx0,_ty0))
endif
```

tx1, ty1, tz1 [tx1_, ty1_, tz1_, _tx1, _ty1, _tz1]

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Для команд движения параметры равны выраженным в **дискретах** проекциям на координатные оси вектора, касательного траектории движения в конечной точке анализируемого [предшествующего, последующего] движения. Длина этого вектора принимается равной длине перемещения для линейных перемещений (в этом случае txl = dx, tzi = dz, tzi = dz) и радиусу дуги для перемещений по окружности. Направление касательного вектора совпадает с направлением движения в конечной точке перемещения. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой движения.

Параметры изменения направления траектории инструмента

qf0 [qf0_, _qf0]

Тип

вещественный

Умолчание

0.

Описание

Для команд движения параметр равен углу поворота между проекциями направлений траектории движения в начальной точке анализируемого [предшествующего, последующего] движения и в конечной точке предшествующего движения на плоскость интерполяции предшествующего движения. Если анализируемая [предшествующая, последующая] команда исходной программы является первой командой движения в главной программе, подпрограмме или проходе с заданным инструментом, параметр принимает нулевое значение. Угол считается положительным, если траектория поворачивается по часовой стрелке, и отрицательным в противном случае. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой движения.

qf1 [qf1_, _qf1]

Тип

вещественный

Умолчание

0.

Описание

Для команд движения параметр равен углу поворота между проекциями направлений траектории движения в конечной точке анализируемого [предшествующего, последующего] движения и в начальной точке последующего движения на плоскость интерполяции анализируемого [предшествующего, последующего] движения. Если анализируемая [предшествующая, последующая] команда исходной программы является последней командой движения в главной программе, подпрограмме или проходе с заданным инструментом, параметр принимает нулевое значение. Угол считается положительным, если траектория поворачивается по часовой стрелке, и отрицательным в противном случае. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой движения.

---

path_rot_0# [path_rot_0_#, _path_rot_0#]

Тип

перечислимый: PATH_BEGIN#, PATH_END#, PATH_BREAK#, PATH_LEFTROT#, PATH_RIGHTROT#, PATH_NULLROT#

Умолчание

не определено

Описание

Для команд движения значение параметра соответствует характеру сопряжения траектории перемещения в начальной точке анализируемого [предшествующего, последующего] движения. Принимает значения перечислимого типа, имеющие следующий смысл:

PATH_BEGIN# - анализируемая [предшествующая, последующая] команда исходной программы является первой командой движения в главной программе, подпрограмме или проходе с заданным инструментом;

PATH_BREAK# - излом траектории, не лежащий в текущей плоскости интерполяции;

PATH_LEFTROT# - "левый" излом траектории в плоскости интерполяции данного движения;

PATH_RIGHTROT# - "правый" излом траектории в плоскости интерполяции данного движения

PATH_NULLROT# - гладкое сопряжение траектории.

Для команд, не являющихся командами перемещения, значение параметра не определено.

path_rot_1# [path_rot_1_#, _path_rot_1#]

Тип

перечислимый: PATH_BEGIN#, PATH.END, PATHJ3REAK, PATH_LEFT-ROT#, PATH_RIGHTROT#, PATH_NULLROT#

Умолчание

не определено

Описание

Для команд движения значение параметра соответствует характеру сопряжения траектории перемещения в конечной точке анализируемого [предшествующего, последующего] движения. Принимает значения перечислимого типа , имеющие следующий смысл:

PATH_END# - анализируемая [предшествующая, последующая] команда исходной программы является последней командой движения в главной программе, подпрограмме или проходе с заданным инструментом

PATH_BREAK# - излом траектории, не лежащий в текущей плоскости интерполяции

PATH_LEFTROT# - "левый" излом траектории в плоскости интерполяции данного движения;

PATH_RIGHTROT# - "правый" излом траектории в плоскости интерполяции данного движения;

PATH_NULLROT# - гладкое сопряжение траектории.

Для команд, не являющихся командами перемещения, значение параметра не определено.

Координаты исходной точки траектории

x0, y0, z0 [x0_, y0_, z0_, _x0, _y0, _z0]

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Заданные в дискретах координаты точки отсчета для анализируемой [предшествующей, последующей] команды исходной программы или подпрограммы. Для команды заданной параметром SET_ORIGIN# значения параметров есть смещения по координатным осям текущей точки исходной программы относительно нового начала координат.

Сохраняются неизменными до прихода новой команды с кодом SET_ORIGIN#. Дискретность этих параметров задается величиной RDISCR.

Пример

```
elseif command# = SET_ORIGIN# then     ' Вывод информации,
                          ' связанной с установкой сходной
точки
if proc_off# then                       ' Вывод номера текущего
кадра
write id1# using "N%d", n#             ' Увеличиваем счетчик
кадров
n# = n# + NSTEP#
write id1# using "G92"
write id1# using "X%.15g", RDISCR*x0
write id1# using "Y%.15g", RDISCR*y0
write id1# using "Z%.15g", RDISCR*z0
write id1# using "%s", NL$
endif
```

Дополнительные (4-я, 5-я, 6-я) координаты рабочего органа

---

u, v, w [u_, v_, w_, _u, _v, _w]

Тип

вещественный

Умолчание

параметры u0, v0, w0

Описание

Для команд движения параметры равны выраженным в дискретах 4-й, 5-й, 6-й координатам u,v,w конечной точки анализируемого [предшествующего, последующего] движения, в остальных случаях - координатам текущей точки анализируемой [предшествующей, последующей] команды исходной программы. Физический смысл этих параметров в различных системах может быть различным. Дискретность этих параметров задается величиной XDISCR.

---

du, dv, dw [du_, dv_, dw_, _du, _dv, _dw]

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Для команд движения параметры равны выраженным в дискретах разностям

4-й, 5-й, 6-й координат u,v,w конечной и начальной точки анализируемого [предшествующего, последующего] движения. Значения параметров сохраняются неизменными до тех пор, пока не будут переопределены новой командой движения. Физический смысл этих параметров в различных системах может быть различным. Дискретность этих параметров задается величиной XDISCR.

---

impt_u, impt_v, impt_w [impt_u_, impt_v_, impt_w_, _impt_u, _impt_v, _impt_w]

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Выраженные в дискретах суммы импульсов (величина абсолютного физического перемещения) по 4-й,5-й, 6-й координатам u,v,w текущей точки анализируемой [предшествующей, последующей] команды относительно самой первой точки программы или подпрограммы. Дискретность этих параметров задается величиной XDISCR.

---

u0, v0, w0 [u0_, v0_, w0_, _u0, _v0, _w0]

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Заданные в дискретах 4-я, 5-я, 6-я координаты u,v,w точки отсчета для анализируемой [предшествующей, последующей] команды исходной программы или подпрограммы. Для команды с кодом SET_ORIGIN# значения параметров есть смещения по 4-й,5-й, 6-й координатным осям текущей точки исходной программы относительно нового начала координат. Сохраняются неизменными до прихода новой команды с кодом SET_ORIGIN. Физический смысл этих параметров в различных системах может быть различным. Дискретность этих параметров задается величиной XDISCR.

Движение по второй управляющей плоскости (4D - электроэрозионная обработка).

Смысловое значение параметров, представленых в этом разделе, соответствует значению аналогичных параметров с нижней плоскости (плоскости по умолчанию) того же типа и с теми же значениями по

умолчанию. Синтаксически отличаются от них добавлением префикса up_. Ниже представлена таблица соответствия этих параметров:

Параметры первой плоскости	Параметры второй плоскости
Primitive# [primitive_#, _primitive#]	up_primitive# [up_primitive_#, up_primitive#]
x, y, z [x_, y_, z_, _x, _y, _z]	up_x, up_ y, up_ z [up_x_, up_y_, up_z_, _ up_x, _ up_y, _ up_z]
dx, dy, dz [dx_ ,dy_ ,dz_, _dx, _dy, _dz]	up_dx, up_dy, up_dz [up_dx_ , up_dy_ , up_dz_, _ up_dx, _ up_dy, _ up_dz]
i, j, k [i_, j_, k_, _i, _j, _k]	up_i, up_j, up_k [up_i_, up_j_, up_k_, _ up_i, _ up_j, _ up_k]
di, dj, dk [di_, dj_, dk_, _di, _dj, _dk]	up_di, up_dj, up_dk [up_di_, up_dj_, up_dk_, _ up_di, _ up_dj, _ up_dk]
r [r_, _r]	up_r [up_r_, _ up_r]
tx0, ty0, tz0 [tx0_, ty0_, tz0_, _tx0, _ty0, _tz0]	up_tx0, up_ty0, up_tz0 [up_tx0_, up_ty0_, up_tz0_, _ up_tx0, _ up_ty0, _ up_tz0]
txl, tyl, tzl [txl_, tyl_, tzl_, _txl, _tyl, _tzl]	up_txl, up_tyl, up_tzl [up_txl_, up_tyl_, up_tzl_, _ up_txl, _ up_tyl, _ up_tzl]
qf0 [qf0_, _qf0]	up_qf0 [up_qf0_, _ up_qf0]
qf1 [qf1_, _qf1]	up_qf1 [up_qf1_, _ up_qf1]
path_rot_0# [path_rot_0_#, _path_rot_0#]	up_path_rot_0# [up_path_rot_0_#, _ up_path_rot_0#]
path_rot_1# [path_rot_1_#, _path_rot_1#]	up_path_rot_1# [up_path_rot_1_#, _ up_path_rot_1#]
len [len_, _len]	up_len [up_len_, _ up_len]
al_beg [al_beg_, _al_beg]	up_al_beg [up_al_beg_, _ up_al_beg]
al_end [al_end_, _al_end]	up_al_end [up_al_end_, _ up_al_end]
al_delt [al_delt_, _al_delt]	up_al_delt [up_al_delt_, _ up_al_delt]

---

up_r [up_r_, _ up_r]

Тип

вещественный

Умолчание

не определено

Описание

Соответствует параметру r основной плоскости.

Пример программирования обработки на двух плоскостях

```
 ' Задание типа перемещения учитывающего нижнюю и верхнюю
плоскости G101, G102, G103, G201,G202,G203,G301,G302,G303
 if        primitive# = LOCATE#      then
                X_PRIM# = 1
          elseif    primitive# = LINE#        then
                X_PRIM# = 1
          elseif    primitive# = ARC_CLW#     then
                X_PRIM# = 2
          elseif    primitive# = ARC_CCLW#    then
                X_PRIM# = 3
          endif
          if     up_primitive# = LOCATE#     then
                U_PRIM# = 1
          elseif    up_primitive# = LINE#     then
                U_PRIM# = 1
          elseif    up_primitive# = ARC_CLW#     then
                U_PRIM# = 2
          elseif    up_primitive# = ARC_CCLW#    then
                U_PRIM# = 3
          endif
    G_PRIM# = X_PRIM#*100+U_PRIM#
      write id1# using "G%d", G_PRIM#
       'Вывод линейного перемещения (координат точек) на
        'нижней плоскости
    write id1# using "X%.15gY%.15g", x*RDISCR,y*RDISCR
      'Вывод кругового перемещения (координат центра дуги) на
       нижней плоскости
      if(primitive# = ARC_CLW# OR primitive# = ARC_CCLW#) then
      write id1# using "I%.15gJ%.15g", i*RDISCR,j*RDISCR
      endif
      'Вывод линейного перемещения (координат точек) на
       верхней плоскости
      write id1# using "U%.15gV%.15g", up_x*RDISCR,up_y*RDISCR
      'Вывод кругового перемещения (координат центра дуги) на
       верхней плоскости
      if(primitive# = ARC_CLW# OR primitive# = ARC_CCLW#) then
      write id1# using "R390%+gR391%+g",
up_i*RDISCR,up_j*RDISCR
      endif
```

Признаки начала и конца программы

---

prog_off# [prog_off_#, _prog_off#]

Тип

логический

Умолчание

TRUE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая,

последующая] команда не является элементом главной программы, и FALSE# в противном случае.

---

prog_body# [prog_body_#, _prog_body#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является элементом главной программы, и FALSE# в противном случае.

Пример

```
' Если анализируемая команда перемещения выполняется
' в главной программе то текущие координаты выводятся в
' абсолютных координатах
if prog_body# then
' при ненулевом перемещении по оси X :
if dx<>0.0 then
' Выводится новое значение x :
write id1# using "X%.15g", x*RDISCR
endif
```

---

prog_begin# [prog_begin_#, _prog_begin#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является первой командой перемещения в главной программе, и FALSE# в противном случае.

---

prog_end# [prog_end_#, _prog_end#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая,

последующая] команда является последней командой перемещения в главной программе, и FALSE# - в противном случае.

Признаки начала и конца прохода

Под "проходом" здесь и далее будем подразумевать последовательность команд, начинающуюся и оканчивающуюся командами смены инструмента.

---

pass_begin# [pass_begin_#, _pass_begin#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является первой командой перемещения в текущем проходе (последовательности команд, начинающейся и оканчивающейся командами смены инструмента), и FALSE# в противном случае.

---

pass_end# [pass_end_#, _pass_end#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая,последующая] команда является последней командой перемещения в текущем проходе (последовательности команд, начинающейся и оканчивающейся командами смены инструмента), и FALSE# в противном случае.

Признаки начала и конца подпрограммы

---

proc_off# [proc_off_#, _proc_off#]

Тип

логический

Умолчание

TRUE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда не является элементом текущей подпрограммы, и FALSE# в противном случае.

---

proc_body# [proc_body_#, _proc_body#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является элементом подпрограммы, и FALSE# в противном случае.

Пример

```
if proc_body# then
' при ненулевом перемещении по оси X :
if dx<>0.0 then
' Выводится новое значение dx :
write id1# using "X%.15g", dx*RDISCR
endif
```

---

proc_begin# [proc_begin_#, _proc_begin#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является первой командой перемещения в подпрограмме, и FALSE# в противном случае.

Пример

```
if proc_begin# then
 write id1# using "G91"
' Иначе, если анализируемое перемещение принадлежит главной
программе
' и ему предшествовала команда вызова подпрограммы,
' то выводится G90 для возврата в абсолютный режим :
elseif prog_body# AND (command_ = CALL#) then
 write id1# using "G90"
```

```
endif
```

---

proc_end# [proc_end_#, _proc_end#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является последней командой перемещения в подпрограмме, и FALSE# в противном случае.

Признаки обработки контура в режиме коррекции

---

contour_begin# [contour_begin_#, _contour_begin#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является первой командой перемещения по контуру с включенной коррекцией радиуса инструмента, и FALSE# в противном случае.

---

contour_end# [contour_end_#, _contour_end#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является первой командой перемещения по контуру с включенной коррекцией радиуса инструмента, и FALSE# в противном случае.

---

contour_to# [contour_to_#, _contour_to#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является "подходом" к контуру (на нем происходит включение коррекции радиуса инструмента), и FALSE# в противном случае.

---

contour_from# [contour_from_#, _contour_from#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является "отходом" от контура (на нем происходит выключение коррекции радиуса инструмента), и FALSE# в противном случае.

---

contour_on# [contour_on_#, _contour_on#]

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда является командой перемещения по контуру с включенной коррекцией радиуса инструмента, и FALSE# в противном случае.

---

contour_off# [contour_off_#, _contour_off#]

Тип

логический

Умолчание

TRUE#

Описание

Принимает значение TRUE#, если анализируемая [предшествующая, последующая] команда не является командой перемещения по контуру с включенной коррекцией радиуса инструмента, и FALSE# в противном случае.

Режимы коррекции

---

rcomp# [rcomp_#, _rcomp#]

Тип

перечислимый: RCOMP_OFF#, RCOMP_LEFT#, RCOMP_RIGHT#

Умолчание

RCOMP OFF#

Описание

Принимает значения перечислимого типа, имеющие следующий смысл:

RCOMP OFF# - анализируемая [предшествующая, последующая] команда исходной программы является либо командой перемещения при выключенной коррекции на радиус инструмента (при этом параметр contour_off# принимает значение TRUE#), либо командой перемещения, на которой происходит выключение коррекции радиуса инструмента "отходом" от контура, при этом параметр contour_from# принимает значение TRUE#;

RCOMP_LEFT# , RCOMP_RIGHT# - анализируемая [предшествующая, последующая] команда исходной программы является либо командой перемещения при включенной левой или правой коррекции на радиус инструмента соответственно (при этом параметр contour_on# принимает значение TRUE#), либо командой перемещения, на которой происходит включение коррекции радиуса инструмента "подходом" к контуру (при этом параметр contour_to# принимает значение TRUE#).

В остальных случаях значение параметра не определено.

---

cutter_r_crtr# [cutter_r_crtr_#, _cutter_r_crtr#]

Тип

целый

Умолчание

номер инструмента, заданного в исходной программе

Описание

Номер заданного в исходной программе корректора радиуса инструмента, если анализируемая [предшествующая, последующая] команда исходной программы должна выполняться при включенной коррекции на радиус инструмента. При анализе команды смены инструмента значение параметра

становится равным номеру инструмента, задаваемого этой командой. Значение сохраняется неизменным до тех пор, пока не будет переопределено новой командой подхода к контуру или смены инструмента.

---

lcomp# [lcomp_#, _lcomp#]

Тип

перечислимый: LCOMP_OFF#, LCOMP_UP#, LCOMP_DOWN#

Умолчание

LCOMP OFF#

Описание

Принимает значения перечислимого типа, имеющие следующий смысл

LCOMP_OFF# - анализируемая [предшествующая, последующая] команда исходной программы является либо командой перемещения при выключенной коррекции на длину инструмента соответственно, либо командой перемещения, на которой происходит выключение коррекции длины инструмента;

LCOMP_UP#, LCOMP_DOWN# - анализируемая [предшествующая, последующая] команда исходной программы является либо командой перемещения при включенной положительной или отрицательной коррекции на длину инструмента соответственно, либо командой перемещения, на которой происходит включение соответствующего режима коррекции длины инструмента.

В остальных случаях значение параметра не определено.

---

cutter_l_crtr# [cutter_l_crtr_#, _cutter_l_crtr#]

Тип

целый

Умолчание

номер инструмента, заданного в исходной программе

Описание

Номер заданного в исходной программе корректора длины инструмента, если анализируемая [предшествующая, последующая] команда исходной программы должна выполняться при включенной коррекции на длину инструмента. При анализе команды смены инструмента значение параметра становится равным номеру инструмента, задаваемого этой командой. Значение сохраняется неизменным до тех пор, пока не будет переопределено новой командой перемещения с включением коррекции на длину инструмента или смены инструмента.

Пример

```
Если на анализируемом перемещении изменился режим коррекции
длины инструмента (т.е. параметр lcomp анализируемой команды
не равен аналогичному параметру предшествующей команды
lcomp_#):
if (lcomp<>1cотр_) -then
' Для включения положительной коррекции
' вывод G43 с корректором
' по адресу D :
if lcomp = LCOMP_UP# then
write id1# using "G43D%.2d", cutter_l_crtr#
' Иначе, для включения отрицательной коррекции
' вывод G44 с корректором ' по адресу D :
elseif lcomp = LCOMP_DOWN# then
write id1# using "G44D%.2d", cutter_l_crtr#
' Иначе, для выключения коррекции на длину инструмента
' вывод G40 :
elseif lcomp = LCOMP_OFF# then
 write id1# using "G40"
endif
endif
```

Номер вызываемой подпрограммы

---

unit_call# [unit_call_#, _unit_call#]

Тип

целый

Умолчание

0

Описание

Равен номеру вызываемой подпрограммы, когда анализируемая [предшествующая, последующая] команда исходной программы есть команда вызова подпрограммы (т.е. параметр command# принимает значение CALL#). Сохраняется неизменным до прихода новой команды с кодом CALL.

Пример

```
if command# = CALL# then
 ' Вывод информации,
' связанной с вызовом подпрограммы
'Вывод кадра, содержащего
'номер кадра n# по адресу N
'функцию G22
'номер вызываемой подпрограммы по адресу P
'ограничитель кадра LF :
write id1# using "N%dG22P%d%s", n#, unit_call# , NL$
n# = n# + NSTEP#
endif
```

Номер программы и комментарий (текст на ленте)

partno#

Тип

целый

Умолчание

0

Описание

Равен номеру программы, заданному в исходной программе. Если номер в исходной программе не задан, то равен 0.

Пример

```
if command# = HEADER# then ' Вывод информации в начале
программы.
' Вывод заголовка программы :
write id1# using "PROGRAMME %.2d %-18sM RWED
4",partno#,"CC100M"
write id1#, NL$
endif
```

partno_text$

Тип

строковый

Умолчание

0

Описание

Текст комментария, заданный в заголовке исходной программы пустая строка (строка нулевой длины), если отсутствует в исходной программе.

Пример

```
if command# = HEADER# then      ' Вывод информации в начале
программы.
write id1# using "@%s%s", partno_text$ , NL$
                                  ' Вывод строки PARTNO
                                  '(текст на ленте)
                                  ' предваренной символом '@'
n# = NSTART#                     ' Установка начального
значения
                                  ' счетчика кадров
write id1# using "%%"           ' Вывод знака процента в
начале
                                  ' программы
write id1# using "%s", NL$      ' Вывод знака LF
endif
```

unit_count#

Тип

целый

Умолчание

0

Описание

Количество модулей (главная программа + подпрограммы) в исходной программе. Если присутствует только главная программа, равен 1 иначе равен 1 + число подпрограмм в исходной программе.

---

proc_no#

Тип

целый

Умолчание

0

Описание

Равен номеру подпрограммы, которой принадлежит анализируемая команда исходной программы. Если анализируемая команда принадлежит главной программе, параметр принимает значение 0.

Пример

```
if command# = PROCEDURE# then  ' Вывод информации,
                                     ' связанной с
                                     ' началом подпрограммы.
' Вывод кадра, содержащего
' номер кадра n# по адресу N,
' номер подпрограммы,
' ограничитель кадра LF :
write id1# using "N%d$%d%s", n#, proc_no# , NL$
n# = n# + NSTEP#
elseif command# = RETURN# then      ' Вывод информации,
                                     ' связанной с
                                     ' возвратом из
подпрограммы.
' Вывод кадра, содержащего
' номер кадра n# по адресу N,
' функцию G99,
' ограничитель кадра LF:
write id1# using "N%dG99%s", n# , NL$
n# = n# + NSTEP#
endif
```

Нумерация и очередность проходов

pass_count#

Тип

целый

Умолчание

1

Описание

Если в программе нет команд смены инструмента, равен 1 иначе равен 1 + число смен инструмента.

---

pass_no#

Тип

целый

Умолчание

0

Описание

Номер прохода (последовательности команд, начинающейся и оканчивающейся командами смены инструмента), которому принадлежит анализируемая команда исходной программы. До появления первой команды смены инструмента равен 0. Увеличивается на единицу при анализе очередной команды загрузки инструмента.

Пример

```
if pass_no# > 1 then            ' Если номер прохода больше 1
write id1# using "N%.3dM01FOOOO%s", n# , NL$
                                  ' Выводится кадр с
техостановом M01
n# = n# + NSTEP#                ' Увеличиваем счетчик кадров
endif
```

---

pass_cutter_no#

Тип

целый

Умолчание

0

Описание

Номер инструмента, загруженного в данном проходе (последовательности команд, начинающейся и оканчивающейся командами смены инструмента), которому принадлежит анализируемая команда исходной

программы. До появления первой команды смены инструмента равен заданному в разделе инициализации. Затем становится равным номеру инструмента - параметру команды смены инструмента, с которой начинается текущий проход.

Пример

```
if command# = LOAD_TOOL# then      ' Вывод информации,
                                   ' связанной со сменой
инструмента.
' Вывод кадра, содержащего
' номер кадра n# по адресу N
' двузначный номер инструмента по адресу T
' функцию с M6
' ограничитель кадра LF :
write id1# using "N%dT%.2d%.2dM6%s", n#, pass_cutter_no#,
pass_cutter_no# , NL$
n# = n# + NSTEP#
endif
```

---

next_pass_no#

Тип

целый

Умолчание

0

Описание

Номер следующего прохода (последовательности команд, начинающейся и оканчивающейся командами смены инструмента). Если проход, которому принадлежит анализируемая команда исходной программы - последний, то next_pass_no# = pass_no# иначе, next_pass_no# = pass_no# + 1.

---

prev_pass_no#

Тип

целый

Умолчание

0

Описание

Номер предыдущего прохода (последовательности команд, начинающейся и оканчивающейся командами смены инструмента). Если проход, которому принадлежит анализируемая команда исходной программы - нулевой, то prev_pass_no# = 0 иначе, prev_pass_no# = pass_no# - 1.

---

nxt_pas_cut_no#

Тип

целый

Умолчание

0

Описание

Номер инструмента, загружаемого в следующем проходе (последовательности команд, начинающейся и оканчивающейся командами смены инструмента). Если проход, которому принадлежит анализируемая команда исходной программы - последний, параметр равен номеру инструмента, используемого в текущем проходе. Иначе равен номеру инструмента - параметру команды смены инструмента, с которой начинается следующий проход.

---

prv_pas_cut_no#

Тип

целый

Умолчание

0

Описание

Номер инструмента, загруженного в предыдущем проходе (последовательности команд, начинающейся и оканчивающейся командами смены инструмента). Если проход, которому принадлежит анализируемая команда исходной программы - нулевой или первый, параметр равен номеру инструмента, заданному в разделе инициализации. Иначе равен номеру инструмента - параметру команды смены инструмента, с которой начинается предыдущий проход.

---

fst_pas_cut_no#

Тип

целый

Умолчание

0

Описание

Номер инструмента, загруженного в первом проходе (последовательности команд, начинающейся и оканчивающейся командами смены инструмента).

---

pass_first#

Тип

логический

Умолчание

не определено

Описание

Принимает значение TRUE#, если проход (последовательности команд, начинающейся и оканчивающейся командами смены инструмента), которому принадлежит анализируемая команда исходной программы - первый (только одна команда смены инструмента была обработана интерпретатором), и FALSE# в противном случае.

---

pass_last#

Тип

логический

Умолчание

не определено

Описание

Принимает значение TRUE#, если проход (последовательности команд, начинающейся и оканчивающейся командами смены инструмента), которому принадлежит анализируемая команда исходной программы - последний (при дальнейшей работе ни одной команды смены инструмента не будет обработано интерпретатором), и FALSE# в противном случае.

Пример

```
' Если данное перемещение есть окончание прохода
' причем данный проход не является последним,
' выводится функция M11:
if pass_end# AND ~pass_last# then
 write id1# using "M11"
' Иначе если данное перемещение есть начало прохода
' причем данный проход не является первым,
' выводится функция M10:
elseif pass_begin# AND ~pass_first# then
 write id1# using "M10"
endif
```

Дополнительные признаки обработки контуров с коррекцией

---

contour_to_l

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "подходом" к контуру с включением левой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_to_r#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "подходом" к контуру с включением правой коррекции радиуса инструмента, и FALSE# в противном случае.

Пример

```
' При подходе к контуру с включением левой коррекции
' выводится G41 и двузначный номер корректора :
if contour_to_l# then
 write id1# using "G41T%.2d", cutter_r_crtr#
' Иначе, при подходе к контуру с включением правой коррекции
' выводится G42 и двузначный номер корректора :
elseif contour_to_r# then
 write id1# using "G42T%.2d", cutter_r_crtr#
' Иначе, при отходе от контура с выключением правой коррекции
' выводится G40 :
elseif contour_from# then
 write id1# using "G40"
endif
```

---

contour_to_ll#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "подходом" к контуру "с левой стороны", т.е с поворотом траектории движения инструмента в начальной точке контура относительно направления подхода против часовой стрелки и с включением левой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_to_lr#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "подходом" к контуру "с правой стороны", т.е с поворотом траектории движения инструмента в начальной точке контура относительно направления подхода по часовой стрелке и с включением левой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_to_rr#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "подходом" к контуру "с правой стороны", т.е с поворотом траектории движения инструмента в начальной точке контура относительно направления подхода по часовой стрелке и с включением правой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_to_rl#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "подходом" к контуру "с левой стороны", т.е с поворотом траектории движения инструмента в начальной точке контура относительно направления подхода против часовой стрелки и с включением правой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_to_int#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "подходом" к контуру "с внутренней стороны", т.е с поворотом траектории движения инструмента в начальной точке контура относительно направления подхода в ту же сторону, что и направление включаемой коррекции радиуса инструмента, и FALSE# в противном случае. Другими словами, contour_to_int# = contour_to_ll# OR contour_to_rr#

---

contour_to_ext#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "подходом" к контуру "с наружной стороны", т.е с поворотом траектории движения инструмента в начальной точке контура относительно направления подхода в сторону, противоположную направлению включаемой коррекции радиуса инструмента, и FALSE# в противном случае. Другими словами, contour_to_ext#=contour_to_lr# OR contour_to_rl#

---

contour_from_l#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "отход" от контура с выключением левой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_from_r#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE#

является "отход" от контура с выключением правой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_from_ll#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "отход" от контура "с левой стороны", т.е с поворотом траектории движения инструмента в конечной точке контура относительно собственного направления против часовой стрелки и с выключением левой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_from_lr#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "отход" от контура "с правой стороны", т.е с поворотом траектории движения инструмента в конечной точке контура относительно собственного направления по часовой стрелке и с выключением левой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_from_rr#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "отход" от контура "с правой стороны", т.е с поворотом траектории движения инструмента в конечной точке контура относительно собственного направления по часовой стрелке и с выключением правой коррекции радиуса инструмента, и FALSE# в противном случае.

contour_from_rl#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "отход" от контура "с левой стороны", т.е с поворотом траектории движения инструмента в конечной точке контура относительно собственного направления против часовой стрелки и с выключением правой коррекции радиуса инструмента, и FALSE# в противном случае.

---

contour_from_in#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "отход" от контура "с внутренней стороны", т.е с поворотом траектории движения инструмента в конечной точке контура относительно собственного направления в ту же сторону, что и направление выключаемой коррекции радиуса инструмента, и FALSE# в противном случае. Другими словами, contour_from_in# = contour_from_ll# OR contour_from_rr#

---

contour_from_ex#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда MOVE# является "отход" от контура "с наружной стороны", т.е с поворотом траектории движения инструмента в конечной точке контура относительно собственного направления в сторону, противоположную направлению выключаемой коррекции радиуса инструмента, и FALSE# в противном случае. Другими словами, contour_from_in# = contour_from_lr# OR contour_from_ex#

contour_on_int#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда является командой движения по окружности, и на которой согласно исходной программе направление коррекции радиуса инструмента задаётся от центра этой окружности, и FALSE# в противном случае. Другими словами, "контур обходится по внешней стороне".

---

contour_on_ext#

Тип

логический

Умолчание

FALSE#

Описание

Принимает значение TRUE#, если анализируемая команда является командой движения по окружности, и на которой согласно исходной программе направление коррекции радиуса инструмента задаётся к центру этой окружности, и FALSE# в противном случае. Другими словами, "контур обходится по внутренней стороне".

Пример

```
' При подходе к контуру с внутренней стороны
' выводится L8 и двузначный номер корректора :
if contour_to_int# then
 write id1# using "L8%.2d", cutter_r_crtr#
' Иначе, при подходе к контуру с наружной стороны
' выводится L0 и двузначный номер корректора :
elseif contour_to_ext# then
 write id1# using "L0%.2d", cutter_r_crtr#
' Иначе, при отходе от контура с внутренней стороны
' выводится L8 и двузначный номер корректора :
elseif contour_from_in# then
 write id1# using "L8%.2d", cutter_r_crtr_#
' Иначе, при отходе от контура с наружной стороны
' выводится L0 и двузначный номер корректора :
elseif contour_from_ex# then
 write id1# using "L0%.2d", cutter_r_crtr_#
' Иначе, при обходе контура с внутренней стороны
' выводится L8 и двузначный номер корректора :
elseif contour_on_int# then
 write id1# using "L8%.2d", cutter_r_crtr#
```

```
' Иначе, при обходе контура с наружной стороны
' выводится L0 и двузначный номер корректора :
elseif contour_on_ext# then
 write id1# using "L0%.2d", cutter_r_crtr#
' Иначе, при изменении (включении или выключении)
' режима коррекции длины инструмента
' выводится L4 и двузначный номер корректора :
elseif lcomp<>lcomp_# then
 write id1# using "L4%.2d", cutter_l_crtr#
endif
```

Параметры постоянного цикла

---

cycle_start#

Тип

логический

Умолчание

FALSE

Описание

Параметр определяет, что анализируемая команда непосредственно следует за заданием постоянного цикла. Командами, связанными с постоянным циклом, могут быть MOVE# или CALL#.

Принимает значение TRUE# на первой команде MOVE# или CALL# после определения цикла (команда CYCLE#).

---

cycle_on# [cycle_on_#, _cycle_on#]

Тип

логический

Умолчание

FALSE

Описание

Параметр определяет, что анализируемая команда является командой постоянного цикла. Такими командами могут быть MOVE# или CALL#.

Принимает значение TRUE# на всех командах MOVE# или CALL# после определения цикла и до его отмены.

---

cycle# [cycle_#, _cycle#]

Тип

целый

Умолчание

0

Описание

Номер заданного в исходной программе постоянного цикла для анализируемой [предшествующей, последующей] команды исходной программы. При анализе команды установки параметров постоянного цикла в исходной программе значение параметра становится равным номеру постоянного цикла, задаваемого этой командой. Значение сохраняется неизменным до тех пор, пока не будет переопределено новой командой установки параметров постоянного цикла в исходной программе.

---

cycle_p0, ... cycle_p11 [cycle_p0_, ... cycle_p11_, _cycle_p0, ... _cycle_p11]

Тип

вещественный

Умолчание

0.,.....0.

Описание

Параметры с номерами от 0 до 11 заданного в исходной программе постоянного цикла для анализируемой [предшествующей, последующей] команды исходной программы. При анализе команды установки параметров постоянного цикла в исходной программе значение параметра становится равным параметру постоянного цикла, задаваемого этой командой. Значение сохраняется неизменным до тех пор, пока не будет переопределено новой командой установки параметров постоянного цикла в исходной программе.

В системе ГеММа-3D принято следующее задание параметров для основных циклов фрезерной обработки:

cycle_p0 - рабочая подача

cycle_p1 - высота безопасности

cycle_p2 - высота начала обработки

cycle_p3 - толщина снимаемого слоя

cycle_p4 - задержка

cycle_p5 - обороты шпинделя

cycle_p6 - 98/99 (G-функция) (переход на рабочей плоскости или плоскости перехода)

cycle_p7 - высота перехода между блоками

Прямая вставка текста

literal$ [literal_$, _literal$]

Тип

строка

Умолчание

0

Описание

Символьная строка, предназначенная для непосредственной вставки в выходную программу при обработке анализируемой [предшествующей, последующей] команды исходной программы. Значение сохраняется неизменным до тех пор, пока не будет переопределено в исходной программе.

```
If (Len(literal$) > 1) then
 write id1# using "N%2d",n#
 n# = n# + NSTEP#
' прямая вставка текста
 write id1# using " %s",literal$
 write id1#, NL$
```

Дополнительные технологические параметры

xxm , yym , zzm

Тип

вещественный

Умолчание

0., 0., 0.

Описание

Параметры определяют положение нуля программы или подпрограммы в системе координат детали. Данные необходимы для правильной прорисовки программы.

bxmin , bymin , bzmin , bxmax , bymax , bzmax

Тип

вещественный

Умолчание

0., 0., 0., 0., 0., 0.

Описание

Параметры определяют размер прямоугольной заготовки (или рабочей области), заданной в команде BLOCK языка APT.

---

origno# [origno_#, _origno#]

тип

целый

Умолчание

0

Описание

Определяет номер текущей станочной системы координат. Задается командой ORIGNO языка APT. Действует до следующей команды ORIGNO, изменяющей этот номер.

---

td#

Тип

целый

Умолчание

0

Описание

Заданное в секундах время (DELAY#) для команды останова. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой останова на заданное время.

---

mode#

Тип

перечислимый

MILL#, TURN#, EDM, PUNCH#, NIBBLING#, CUTTING#, CORR_5D#, MILL_5D#

Умолчание

0

Описание

Режим обработки, задаваемый APT командой MODE. Параметр принимает значение MILL# при фрезерной обработке и значение TURN# при токарной обработке. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой режима обработки.

---

axis# (axis_#, _axis#)

Тип

целый

Умолчание

0

Описание

Ось поворота стола, задаваемая анализируемой [предшествующей, последующей] командой ROTATE. Параметр принимает значение A_AXIS# для оси X, B_AXIS# для оси Y, C_AXIS# для оси Z при токарной обработке. Значение параметра сохраняется неизменным до тех пор, пока не будет переопределено новой командой режима обработки.

---

aang (aang_, _aang)

bang (bang_, _bang)

cang (cang_, _cang)

Тип

вещественный

Умолчание

0

Описание

Угол поворота стола, задаваемый анализируемой [предшествующей, последующей] командой ROTATE. Параметр aang дает угол поворота вокруг оси X, bang дает угол поворота вокруг оси Y, cang дает угол поворота вокруг оси Z. Значение параметров сохраняется неизменным до тех пор, пока не будет переопределено новой командой режима обработки.

---

flush#

Тип

целый

Умолчание

0

Описание

Параметр промывки, задаваемый при электроэрозионной резке.

Команда APT

FLUSH/n, где n -параметр промывки

---

genrtr#

Тип

целый

Умолчание

0

Описание

Параметр генератора, задаваемый при электроэрозионной резке.

Команда APT

GENRTR/n, где n -параметр генератора

---

edm_angle

Тип

вещественный

Умолчание

0

Описание

Параметр определяет величину угла наклона проволоки при 2D электроэрозионной резке.

Команда APT

EDM_ANGLE/ угол наклона проволоки, номер углового корректора

---

edm_ang_corr#

Тип

целый

Умолчание

0

Описание

Параметр определяет номер углового корректора проволоки при 2D электроэрозионной резке.

Команда APT

EDM_ANGLE/ угол наклона проволоки, номер углового корректора

---

zplane (zplane _, _ zplane)

тип

вещественный

Умолчание

0

Описание

Координата нижней управляющей плоскости в милиметрах при 4D электроэрозионной резке.

Команда APT

Z_PLANE/ Z нижней плоскости , Z верхней плоскости

---

up_zplane (up_zplane _, _ up_zplane)

тип

вещественный

Умолчание

0

Описание

Положение верхней управляющей плоскости (задается в милиметрах) при 4D электроэрозионной резке.

Команда APT

Z_PLANE/ Z нижней плоскости , Z верхней плоскости

---

punch# (punch_#,_punch#)

Тип

целый

Умолчание

0

Описание

Параметр задает число ударов на линейном или круговом перемещении при пробивке.

Команда APT

PUNCH/n, где n - число ударов

nib_step (nib_step _, _ nib_step)

Тип

вещественный

Умолчание

0

Описание

Параметр задает шаг высечки при использовании соответствующего режима пробивки на высечном прессе.

Команда APT

MODE/ { NIBBLING} , шаг высечки