CNC 8

СЧПУ на базе промышленного компьютера.
Параметрическое программирование.

ООО «Новые Электронные Технологии»
г. Азов

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Содержание

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Введение

Для системы числового управления CNC 8 (далее СЧПУ CNC 8) используется язык параметрического программирования, который представляет собой расширение языка программирования управляющих программ. Язык параметрического программирования делает возможным вариантность вычисления, применение логических операторов, работа с проходами инструмента, движениями манипуляторов. Возможность организации циклов, выбор по условию, переход, работа с подпрограммами. Добавляются элементы, осуществляющие полный контроль над СЧПУ CNC 8, — доступ к системным переменным и ячейкам программы электроавтоматики, возможность создавать свои собственные G-коды и функции, которые наиболее полно реализуют управление всех компонентов станка. Возможен доступ к параметрам СЧПУ CNC 8, хранящим информацию об инструменте, положении рабочих органов, манипуляторов, системы координат, значений G-кода управляющей программы и ошибок. С помощью параметрического программирования можно разрабатывать диалоговые управляющие программы, организовать дополнительные информационные окна, систему слежения за дополнительными параметрами, режимы контроля и протоколирования процессов обработки и т.д

Синтаксис языка параметрического программирования

В языке параметрического программирования приняты следующие соглашения: все переменные, операторы, функции и процедуры языка параметрического программирования записываются только строчными буквами латинского алфавита, цифрами и символом подчеркивания «_»;

- длина имени переменной не должна превышать 255 символов;
- имя переменной не может начинаться с цифры;
- переменные, используемые в программе являются локальными и действуют только в пределах текущего файла;
- глобальные переменные текущей программы (действующие в пределах программы и подпрограмм) должны начинаться с символа подчеркивания « _ »; переменные-массивы являются глобальными; в одной строке допускается размещать только один оператор; выполняемая программа может иметь только одно окно вывода; выполняемая программа может иметь только одно диалоговое окно; вывод информации в неактивное окно (находящееся на заднем плане) не производится; номер метки программируется с буквой "N";
- адрес перехода в операторах goto, gosub программируется только числом, без символа "N";
- оператор gosub вызова подпрограммы работает только в пределах текущего файла.

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Ключевые слова

Ключевые слова являются составной частью языка параметрического программирования и их нельзя использовать в качестве имени переменной.

Список зарезервированных слов:

abs	and	arc	arccos	arcsin
arctan	bar	bcolor	circle	close
closed ialog	closewindow	cls	color	compare
cos	dump	ellipse	else	end
endiff	eof	endcont	exp	for
frac	getdata	getdatacadr	getdatage	geteadata
getparameter	getstring	getsystemdata	gettime	gosub
goto	if	iff	input	in
locate	msg	next	not	open
or	paramactive	pos	print	or
paramactive	pos	print	pset	read
rect	rem	restore	return	round
seteadata	setsystemdata	sin	sqrt	str
tan	then	to	trunc	val
window	write	xor		

Операции языка параметрического программирования

Арифметические	
+	сложение аргументов
-	вычитание или унарный минус
*	умножение
/	деление
\	остаток от деления (деление по модулю)
=	присвоение или сравнение на равенство
( )	скобки, изменение приоритета операций
<	сравнение на "меньше"
>	сравнение на "больше"
^	возведение в степень
Логические	

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

and	логическое умножение
or	логическое сложение
not	логическое отрицание (логическая инверсия)

Выполнение арифметических и логические операции производится в порядке их приоритетов.

приоритеты операций в порядке возрастания	низший приоритет «+», «-», «or», «xor» «*», «/», «\», «and»;«^»
	унарный минус(плюс), «not»
	высший приоритет «(», «)»

Операторы

Операторы это элемент языка, задающий полное описание действия, которое необходимо выполнить. Каждый оператор представляет собой законченную фразу языка программирования и определяет некоторый вполне законченный этап обработки данных. В состав операторов могут входить служебные слова, данные, выражения и другие операторы, которые могут активизировать процедуру или функцию или передавать управление на другую часть кода.

примеры операторов	c = a + b	(присвоить значение)
	window(20,20,200,300)	(активизировать процедуру)
	if x < 2 then goto 100	(оператор условия)
	for i=1 to 100	(оператор цикла)
	x=i*2	(присвоить значение)
	next	(оператор ограничения цикла)

Выражения

Оператор языка параметрического программирования состоит из выражений. Выражением называется совокупность переменных, констант, знаков операций, имен функций, скобок, которая может быть вычислена в соответствии с синтаксисом языка программирования. Результатом вычисления выражения является величина определенного типа.

примеры выражений	x + y
	done < > error
	i < = length -x
	a[i + 20

Лексемы

Лексема в программе языка параметрического программирования – последовательность машинных символов исходного кода программы, имеющие определенное совокупное

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

значение. Лексемы зарезервированные слова, идентификаторы, числовые константы (целые и действительные числа), буквенные константы, строчные константы, коды операторов, комментарии.

примеры лексем	print	(зарезервированное слово)
	(	(специальный символ)
	=	(специальный символ)
	9	(число)
	"Это текст"	(строковая константа)

Типы, переменные, константы

Константа, переменная — это основополагающие понятия в языке параметрического программирования. Константа — это величина, которая при выполнении программы остаётся неизменной. Переменная — это ячейка памяти для временного хранения данных. В процессе выполнения программы значения переменных могут изменяться.
Каждая переменная может быть только одним из двух типов: вещественный или строковый. Признаком строковой переменной является наличие символа $ в конце имени. Тип переменной определяет множество значений, которые может иметь переменная.
Например, в следующей программе используются переменные x и y, имеющие вещественный тип. Таким образом, x и y могут содержать только числа. Программа остановится из за возникновения ошибки в случае попытки присвоить этим переменным значения строкового типа.

примеры переменных	y = 7.5	(переменной y присваивается значение)
	x = 44	(переменной x присваивается значение)
	y = y + x	(значение переменной x изменяется)

Переменной y в этой программе первоначально присваивается значение 7.5; двумя операторами ниже ей присваивается новое значение: y + x. Значение переменной может изменяться.

Можно использовать массивы в качестве переменных, или упорядоченные множества переменных. Доступ к элементам массивов осуществляется через их индексы. Перед использованием массива он должен быть описан с помощью оператора dim

```
dim a[5]
```

В данном случае описан массив с именем a и с количеством элементов 5.
Индексы массива, указываемые в квадратных скобках, начинаются с нуля. Запись значения 10 в третий элемент массива a будет выглядеть так:

```
a[2]=10
```

Массивы могут быть одномерными (см. выше), так и многомерными. Максимальная раз-

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

мерность массива - 6 индексов и количество элементов в массиве не должно превышать 1000000.

пример двумерного массива	dim mm[3,5] a=2 b=4 mm[a,b]=152

Номер кадра (метка)

Номером кадра (меткой) является символ "N" и числовое значение. Нули расположенные перед символом считаются значащими, то есть если в операторе goto указана метка 00123, то переход будет происходить именно на метку N00123, а не на метку N123. Метки используются с операторами перехода goto и вызова подпрограммы gosub.

Справочник по функциям и процедурам макроязыка

Описание всех функций языка.

Алгебраические и тригонометрические функции

Функция **abs**

Функция	Возвращает абсолютное значение аргумента.
Описание	abs(x)
Окно вывода	2.3 157
Пример	r=abs(-2.3) i = abs(-157) window(100,100,200,200) print r rint i end

Функция **arctan**

Функция	Возвращает арктангенс аргумента
Описание	arctan(x)
Окно вывода	Арктангенс 1 = 45 град
Пример	x=1 r=180/3.14159 * arctan(x) window(100,100,200,200) print "Арктангенс 1 =", r, "град" end
Примечание	Результат представляет собой главное значение арктангенса x (при технологическом параметре N3033, равном 0-в радианах, при N3033=1 - в градусах)

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Функция **tan**

Функция	Возвращает тангенс аргумента
Описание	tan(x)
Окно вывода	Тангенс 45 = 1
Пример	x = 45/180* 3.14159 r = tan(x) window(100,100,200,200) print "Тангенс 45 =", r end
Примечание	Результат представляет собой значение тангенса x, заданного в радианах при N3033=0, в градусах- при N3033=1

Функция **sin**

Функция	Возвращает синус аргумента
Описание	sin(x)
Окно вывода	Синус 30 = 0.
Пример	x = 30*3.14159/180 r = sin (x) window (100,100,200,200) print "Синус 30 = ", r end
Примечание	Результат представляет собой значение синуса x, заданного в радианах при N3033=0, в градусах- при N3033=1

Функция **cos**

Функция	Возвращает косинус аргумента
Описание	*cos(x)*
Окно вывода	Косинус 60 = 0,5
Пример	x = 60*3.14159/180 Г = cos (x) window (100,100,200,200) print "Косинус 60 = ", r end
Примечание	Результат представляет собой значение косинуса x, заданного в радианах при N3033=0, в градусах- при N3033=1

Функция **arcsin**

Функция	Возвращает арксинус аргумента
Описание	arcsin(x)
Окно вывода	Арксинус –0.5= - 30 град
Пример	x=-0.5

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	r=180/3.14159 *arcsin(x) window (100,100,200,200) print "Арксинус -0.5 = ", r ,"град" end
Примечание	Результат представляет собой значение арксинуса x в радианах при N3033=0, в градусах- при N3033=1

Функция **arccos**

Функция	Возвращает арккосинус аргумента
Описание	arccos(x)
Окно вывода	Арккосинус 0.5= 60 град
Пример	x= 0.5 r = 180/3.14159* arccos(x) window (100,100,200,200) print "Арккосинус 0.5 = ", r ,"град" end
Примечание	Результат представляет собой значение арккосинуса x в радианах при N3033=0, в градусах - при N3033=1.

Функция **exp**

Функция	Возвращает экспоненциальное значение аргумента
Описание	exp *(x)*
Окно вывода	e^4 = 54.598
Пример	x = 4 a = exp (x) window (100,100,200,200) print "e^4 = ", a end
Примечание	Параметр x это выражение вещественного типа. Результатом будет экспонента x, то есть значение e возводится в степень x (e - основание натурального логарифма).

Функция **lg**

Функция	Возвращает десятичный логарифм аргумента
Описание	lg (x)
Окно вывода	lg(100) = 2
Пример	x = 100 a = lg(x) window (100,100,200,200) print "lg(100) = ", a end
Примечание	Параметр x является выражением вещественного типа. Результатом будет десятичный логарифм выражения x.

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Функция **ln**

Функция	Возвращает натуральный логарифм аргумента
Описание	ln (x)
Окно вывода	ln(2.718) = 1
Пример	x = 2.718 a = ln(x) window (100,100,200,200) print "ln(2.718) = ", a end
Примечание	Параметр x является выражением вещественного типа. Результатом будет натуральный логарифм выражения x.

Функция **frac**

Функция	Возвращает дробную часть аргумента
Описание	frac(x)
Окно вывода	0,456
Пример	f = frac(123.456) window (100,100,200,200) print f end
Примечание	Параметр x является выражением вещественного типа. Результатом является дробная часть x, то есть frac(x) =x-int(x)

Функция **round**

Функция	Округляет значение вещественного типа до значения целого типа
Описание	round (x)
Окно вывода	123
Пример	r=round(123.456) window (100,100,200,200) print r end
Примечание	Параметр x это выражение вещественного типа. Функция round возвращает значение, которое является значением x, округленным до ближайшего целого числа. Если значение x находится точно посередине между двумя целыми числами, то результатом будет число с большим абсолютным значением.

Функция **int**

Функция	Возвращает целую часть аргумента
Описание	int(x)
Окно вывода	123
Пример	i = int (123.756) window (100,100,200,200)

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	print i end
Примечание	Параметр x это выражение вещественного типа. Результатом будет целая часть x, то есть дробная часть x отбрасывается (округляется в сторону нуля).

Функция **sqrt**

Функция	Возвращает квадратный корень аргумента
Описание	sqrt(x)
Окно вывода	3
Пример	i = 9 window (100,100,200,200) print sqrt(9) end
Примечание	Параметр x это выражение вещественного типа. Результатом будет квадратный корень x.

Функции управления программой

Процедура **end**

Функция	Завершает выполнение программы
Описание	end
Примечание	1. Процедура end может находиться в любом месте программы. Появление этой процедуры останавливает выполнение программы даже в том случае, когда ниже есть какой-либо код или процедура появилась в подпрограмме. 2. Программы, которые отрабатываются в автоматическом режиме, должны останавливаться по командам M2, M30.

Оператор повторения **for next**

Функция	Оператор повторения for разрешает циклически выполнять группу операторов, расположенных ниже до ближайшего оператора next
Описание	for <выражение начала цикла> to <выражение конца цикла>
Пример	Пример: ford=10to124 a=d*2 next Пример: x=12 y = 35 for d = x * 20 to y * 40 for j = 1 to 10 a = j*d+123 next

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	next
Примечание	<Выражение начала цикла> это оператор присвоения управляющей переменной начального значения. <Выражение конца цикла> это либо число, либо выражение, результатом которого является число. В процессе выполнения управляющая переменная увеличивается на 1 в каждом цикле, до тех пор, пока ее значение не превысит значение <Выражения конца цикла>. Операторы цикла могут быть вложенными. Когда начинает выполняться оператор for, начальное и конечное значения определяются один раз, и эти значения сохраняются на протяжении всего выполнения оператора for.

Условный оператор **iff then else endiff**

Функция	В условном операторе, в зависимости от заданных условий выполняется или не выполняется некоторая группа операторов.
Описание	*iff <выражение> then* *<группа операторов> else* *<группа операторов> endiff*
Пример	iff a>b then c=1 b=2 else c=10 b=20 endiff
Ограничение	Параметр x это выражение вещественного типа. Результатом будет квадратный корень x Группа операторов не может содержать более 50 операторов. Нельзя пропускать операторы else и endiff.

Условный оператор **if then**

Функция	В условном операторе, в зависимости от заданных условий выполняется или не выполняется некоторый оператор
Описание	end if <выражение> then <оператор>
Пример	if a>b then c=1 if c=1 then goto 20

Оператор **gosub**

Функция	Осуществляет вызов подпрограммы
Описание	gosubn
Пример	gosub 123 …… end N123 print "Это подпрограмма" Return
Примечание	Параметр n это целое число и определяет номер строки, на которую

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	происходит переход. Номер строки обозначается символом "N" и номером. Ограничения: Подпрограмма должна располагаться обязательно в текущем файле

Оператор **return**

Функция	Осуществляет выход из подпрограммы
Описание	return
Пример	gosub 123 …… end N123 print "Это подпрограмма" Return
Примечание	Выход происходит на последний выполненный оператор вызова подпрограммы gosub. Ограничения: Подпрограмма должна располагаться обязательно в текущем файле

Оператор **goto**

Функция	Вызывает переход на заданную строку
Описание	goto n
Пример	goto 123 N10 print "Текст" goto 500 N123 goto 10 N500 print "TEXT"
Примечание	Параметр n это собой целое число и определяет номер строки, на которую происходит переход. Номер строки обозначается символом "N" и номером.

Оператор **getstring**

Функция	Вызывает чтение очередного кадра, который находится в программе ниже вызова подпрограммы, содержащей оператор getstring
Описание	a$=getstring В строковую переменную a$ пересылается текст следующего кадра.
Пример	N100P50 N101G1X100F200 N102Y100 N103Z100 N104X0Y0Z0 Подпрограмма 50: a$=getstring - в переменной a$ текст кадра N101 b$=getstring - в переменной b$ текст кадра N102 c$=getstring - в переменной c$ текст кадра N103 M99 - возврат в программу на кадр N104
Примечание	оператор может использоваться только в подпрограмме, при этом каждый раз происходит чтение следующего кадра, находящегося в программе, из которой была вызвана данная подпрограмма;

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	после использования оператора точка возврата из подпрограммы по функции M99 сдвигается на следующий кадр; для восстановления точки возврата в исходное состояние необходимо использовать оператор restore.

Оператор **restore**

Функция	Восстанавливает точку возврата из подпрограммы в исходное состояние, то есть на кадр, находящийся после обращения к подпрограмме.
Описание	restore
Пример	Пример: N100P50 N101G1X100F200 N102Y100 N103Z100 N104X0Y0Z0 Подпрограмма 50: a$=getstring - в переменной a$ текст кадра N101 b$=getstring - в переменной b$ текст кадра N102 c$=getstring - в переменной c$ текст кадра N103 restore M99 - возврат в программу на кадр N101
Примечание	оператор может использоваться только в подпрограмме

Оператор **endcont**

Функция	Пустой оператор, не вызывающий никаких действий
Описание	endcont
Пример	N100P50 N101G1X100F200 N102Y100 N103Z100 Endcont N104X0Y0Z0 Подпрограмма 50: N200a$=getstring - в переменной a$ текст кадров N101-N103 If compare(a$,"endcont")<>0 then goto 200 - проверка конца фрагмента программы restore - восстановление точки возврата из подпрограммы M99 - возврат в программу на кадр N101
Примечание	Оператор используется для ограничения фрагмента программы, которая обрабатывается с помощью оператора getstring. При реальной отработке этого фрагмента данный оператор просто пропускается.

Функции графического вывода

Соответствие значений номеров цветов

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Значение	Цвет
0	черный
1	синий
2	зеленый
3	бирюзовый
4	красный
5	малиновый
6	коричневый
7	светло-серый
8	темно-серый
9	светло-голубой
10	светло-зеленый
11	светло-бирюзовый
12	светло-красный
13	светло-малиновый
14	желтый
15	белый

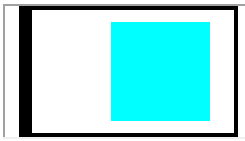
Функции графического вывода

Процедура **arc**

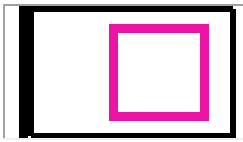
Функция	Вычерчивает дугу окружности от начального угла до конечного угла
Описание	*arc(X,Y,start_angle,end_angle, radiusX, radiusY, colour)*
Пример	wi ndow(20,20,400,400) for r = 1 to 5 arc(100,100, 0, 89, r*10, r*10,1) next end
Окно ввода	 В этом примере рисуется 5 дуг окружности, с радиусами от 10 до 50 с центром в точке 100,100 синим цветом
Примечание	Примечания: Рисует дугу эллипса с центром (x,y) и радиусами *"radiusX"* и *"radiusY".* Дуга рисуется от начального угла *("start_angle")* до конечного угла *("end_angle").* Начальный угол, равный 0 и конечный угол, равный 359, задают вычерчивание полной окружности.
Ограничения	Предварительно должна быть выполнена процедура *window.* Если процедура *window* не выполнена, то процедура *arc будет про*игнорирована и программа продолжит свою работу.

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Процедура **bar**

Функция	Рисует закрашенный прямоугольник
Описание	*bar(x1, y1, x2, y2 ,colour)*
Пример	window(20,20,400,400) X=100 Y=100 bar(x, y,x+200,y+200,3) end В этом примере рисуется закрашенный квадрат со стороной 200 пикселей бирюзовым цветом
Окно ввода	
Примечание	Рисуется закрашенный прямоугольник. Координаты верхнего левого угла задаются точкой *x1* и *y1,* координата правого нижнего угла задаются точкой *x2* и *y2.* Цвет прямоугольника задается параметром *colour*
Ограничения	Сначала должна быть выполнена функция *window.* Если процедура *window* не выполнена, то процедура *bar будет про*игнорирована и программа продолжит свою работу.

Процедура **rect**

Функция	Рисует прямоугольник
Описание	*red (x1, y1, x2, y2, colour)*
Пример	wi ndow(20,20,400,400) x=100 y=100 rect(x, y,x+200,y+200,5) end В этом примере рисуется квадрат со стороной 200 пикселей малиновым цветом.
Окно ввода	
Примечание	Рисуется прямоугольник. Координаты верхнего левого угла задаются точкой *x1* и *y1,* координата правого нижнего угла задаются точкой *x2* и *y2.* Цвет прямоугольника задается параметром *colour.*
Ограничения	Сначала должна быть выполнена функция *window.* Если процедуру *window* не выполнить, то процедура *rect будет про*игнорирована и программа продолжит свою работу.

Процедура **line**

Функция	Рисует линию
Описание	*line (x1, y1, x2, y2, colour)*

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Пример	window(20,20,400,400) x=100 y=100 line (x, y,x+200,y+200,7) end В этом примере рисуется линия от точки (100,100) до точки (300,300) светло-серым цветом.
Окно ввода	
Примечание	Рисуется линия. Координаты верхнего левого угла задаются точкой *x1* и *y1,* координата правого нижнего угла задаются точкой *x2* и *y2.* Цвет линии задается параметром *colour.*
Ограничения	Сначала должна быть выполнена функция *window.* Если процедуру *window* не выполнить, то процедура *line будет про*игнорирована и программа продолжит свою работу.

Процедура **circle**

Функция	Рисует окружность
Описание	red (x, y, radius, colour)
Пример	window(20,20,300,300) x=100 y=100 circle(x, y,75,9) end В данном примере рисуется окружность с центром в точке (100,100) и радиусом 75 светло-голубым цветом.
Окно ввода	
Примечание	Рисуется окружность. Координаты центра задаются точкой x и y, радиус окружности задается параметром radius. Цвет линии задается параметром colour
Ограничения	Сначала должна быть выполнена функция window. Если процедуру window не выполнить, то процедура circle будет проигнорирована и программа продолжит свою работу.

Процедура **ellipse**

Функция	Рисует эллипс
Описание	*ellipse(x1, y1, radiusX, radiusY,colour)*
Пример	window(20,20,300,300)

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	x=100 y=100 ellipse(x, y,100,50,11) end В данном примере рисуется эллипс с центром в точке (100,100) и осью X 100 пикселей и осью Y 50 пикселей светло-бирюзовым цветом.
Окно ввода	
Примечание	Рисуется эллипс. Координаты центра задаются точкой x и y, оси эллипса задаются параметрами *radiusX* и *radiusY.* Цвет эллипса задается параметром *colour.*
Ограничения	Сначала должна быть выполнена функция *window.* Если процедуру *window* не выполнить, то процедура *ellipse будет про*игнорирована и программа продолжит свою работу.

Процедура **pset**

Функция	Рисует точку
Описание	*pset(x, y, colour)*
Пример	window(20,20,400,400) pset(100,100,3)
Окно ввода	
Ограничения	Сначала должна быть выполнена функция window. Если процедура *window* не выполнить, то процедура *pset будет про*игнорирована и программа продолжит свою работу

Процедура **cls**

Функция	Очищает окно вывода программы (закрашивает его цветом, установленным процедурой *bcolor)*
Описание	*cls*
Пример	window(20,20,400,400) bcolor(0) pset(100,100,3)
Окно ввода	
Ограничения	Сначала должна быть выполнена функция window. Если процедуру window не выполнить, то процедура cls будет проигнорирована и программа продолжит свою работу

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Функции текстового вывода

Процедура **print**

Функция	Выводит на экран значение переменных, функций и выражений
Описание	*print a,b,...*
Пример	a=2.3 b=10 c=3 window(20,20,400,400) print a,b*c, "Текст" end
Окно ввода	
Ограничения	Сначала должна быть выполнена функция *window.* Если процедура *window* не выполнена, то процедура *print будет про*игнорирована и программа продолжит свою работу.

Процедура **locate**

Функция	Определяет новое значение точки вывода для оператора *print.*
Описание	*locate (x, y)*
Примечание	Параметры x и y являются координатами относительно левого верхнего угла окна вывода.
Ограничения	Сначала должна быть выполнена функция *window. Процедура locate будет проигнорирована и программа продолжит свою работу* если процедура *window* не будет выполнена

Процедура **msg**

Функция	Выводит в панель индикации сообщений значение переменных, функций, выражений и текстовых строк.
Описание	*msg a,b,...*
Пример	msg "Значение температуры",T

Строковые функции

Функция **concat**

Функция	Сцепляет две строковые переменные
Описание	*concat(b$,"строка2")*
Пример	b$="FMS-" c$="3000" a$ = concat(b$,c$)
Примечание	Каждый параметр это выражение строкового типа. Результат равен объединению двух строковых параметров. При длине результирующей строки

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	превышающей 255 символов, выдается сообщение об ошибке и программа останавливается.

Функция **copy**

Функция	Возвращает для строки подстроку
Описание	*copy(s$, index, count)*
Пример	b$ = 'ABCDEF'; a$ = copy(b$,2,3)
Примечание	Параметр *s$* - выражение строкового типа. Параметры *index* и *count* это выражения целого типа. Функция *copy* возвращает строку, число символов которой соответствует параметру *count* и которая начинается с символа строки *s$,* номер которого задан параметром *index. Пустая строка возвращается е*сли значение параметра *index* превышает длину строки. Возвращается только остаток строки если параметр *count* задает больше символов, чем остается в строке, начиная с символа *index*.

Функция **delete**

Функция	Убирает из строки подстроку
Описание	*delete (s$, index, count)*
Пример	***b$ = 'ABCDEF';*** ***a$ = delete(b$,2,2)***
Примечание	Параметр *s$* это выражение строкового типа. Параметры *index* и *count* это выражения целого типа. Функция *delete* убирает символы, количество которых соответствует параметру *count,* начиная с символа строки *s$,* номер которого задан параметром *index.* Если значение параметра *index* превышает длину строки, то символы не удаляются. Остаток строки удаляется если параметр *count* задает больше символов, чем остается в строке, начиная с символа *index*.

Функция **insert**

Функция	Помещает в строку подстроку
Описание	*insert (s1$, s2$, index)*
Пример	***b$ = 'ABCDEF';*** ***a$ = delete(b$,2,2)***
Примечание	Параметры *s1$, s2$* это выражения строкового типа. Параметр *index* является выражением целого типа. Данная функция вставляет строку, которую задает параметр *s2$,* в строку, задаваемую параметром *s1$,* начиная с позиции, определяемой параметром *index.* В случае когда получившаяся в результате строка превышает 255 символов, то возникает сообщение об ошибке и программа останавливается.

Функция **pos**

Функция	Ищет подстроку в строке
Описание	*pos (sub$, s$)*

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Пример	***s$ = " 123.5"*** ***i = pos("23",s$)***
Примечание	Параметры *sub$* и *s$* это выражения строкового типа. Эта функция ищет подстроку, заданную параметром *sub$,* в строке *s$* и возвращает целое значение, которое является позицией первого символа подстроки в строке *s$. Функция возвращает значение 0 в том случае е*сли подстрока не найдена

Функция **length**

Функция	Возвращает динамическую длину строки
Описание	*length (s$)*
Пример	**s$ = 'a bc';** **i = length(s$)**
Примечание	Параметр *s$* это выражение строкового типа. Результатом будет длина *s$.*

Функция **val**

Функция	Изменяет строковое значение в его численное представление
Описание	val (s$)
Пример	s$="23.456" i=val(s$)
Примечание	Параметр s$ это выражение строкового типа. Функция val изменяет строку s$ в ее численное представление. Если где-либо в строке встречается недопустимый символ, то появляется сообщение об ошибке и программа останавливается.
Ограничения	Предыдущие пробелы нужно удалить

Функция **compare**

Функция	Сопоставляет два строковых значения
Описание	compare (a$, b$)
Пример	if compare (a$, "text") then goto 100
Примечание	Параметры a$ и b$ это выражения строкового типа. Функция compare сопоставляет строки a$ и b$ и возвращает 0, в случае равенства и 1 в случае неравенства.

Функция **copynum**

Функция	Возвращает для строки подстроку, содержащую строковое представление числа
Описание	copynum (s$, index)
Окно ввода	-123.765
Пример	if compare (a$, "text") then goto 100 s$="X-123.765" n$=copynum(s$, 2) window(20, 20, 200, 200) print n$

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	end
Примечание	Параметр s$ - выражение строкового типа. Параметр index это выражение целого типа. Функция copynum возвращает строку, начинающуюся с символа строки s$, номер которого задан параметром index, а из строки s$ копируются только символы "+","-","." и цифры. Строка возвращается пустая при значении параметра index превышающем длину строки.

Функция **str**

Функция	Преобразует число в его символьное представление.
Описание	str *(a:n)*
Пример	a=23.56 b$=str(a:3) Значение переменной b$=23.560 c$=str(a:0) Значение переменной c$=24 d$=str(a) Значение переменной d$=2.35599999999977E+0001
Примечание	Параметр a представляет собой выражение числового типа. Параметр *n* определяет количество знаков после запятой в символьном представлении значения выражения, а также знак, до которого округляется это значение перед преобразованием. Отсутствие параметра *n* соответствует экспоненциальной форме представления значения выражения.

Функции работы с временем и датой

Процедура **getdate**

Функция	Возвращает текущую дату, установленную в операционной системе
Описание	*getdate (year, month, day)*
Пример	Getdate (y, m, d)
Примечание	Возвращаемые значения имеют следующие диапазоны: *year* (год) - 1980..2099, *month* (месяц) -1..12, day (число) 1..31.

Процедура **gettime**

Функция	Возвращает установленное в операционной системе текущее время
Описание	*gettime (hour, min, sec)*
Пример	*Gettime (h, m, s)*
Примечание	Возвращаемые параметры принимают следующие значения: *hour (час)* от 0 до 23, *min* (минута) - от 0 до 59, *sec* (секунда) - от 0 до 59.

Функции работы с окнами и цветом

Процедура **color**

Функция	Определяет цвет, которым выводится текстовая информация
Описание	*color (c)*

Пример	window(1,1,600,400) color(3)
Примечание	Параметр с представляет собой выражение целого типа и определяет код цвета

Процедура **bcolor**

Функция	Определяет цвет фона окна, которым выводится текстовая информация
Описание	*bcolor(c)*
Пример	window(1,1,600,400) bcolor(6)
Примечание	Параметр с представляет собой выражение целого типа и определяет код цвета.

Процедура **window**

Функция	Определяет на экране окно
Описание	*window (x1, y1, x2, y2)*
Пример	*window (x1, y1, x2, y2)*
Примечание	Параметры *x1, y1* это координаты верхнего левого угла окна, параметры *x2, y2* - это координаты правого нижнего угла. Правый левый угол экрана соответствует координате (1,1). Максимальный размер окна определяется как (1,1,800,540). Все координаты, указанные в процедурах (кроме самих координат окна) представляются относительными координатами данного окна. Например, *locate(1,1)* всегда позиционирует курсор на верхний левый угол текущего окна. Все процедуры и функции вывода зависят от текущего окна. Если в программе процедура window уже была выполнена, то следующий вызов приводит к изменению размеров окна и появлении его на переднем плане экрана.

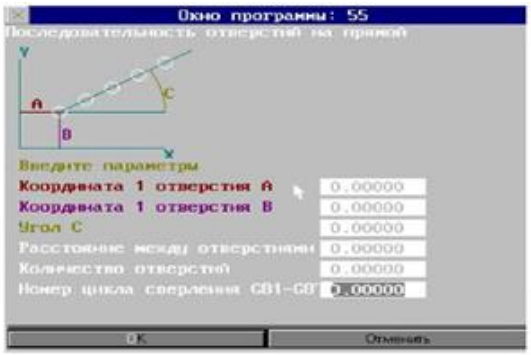
Процедура **closewindow**

Функция	Закрывает окно программы
Описание	*closewindow*
Пример	window(1,1,600,400) closewindow
Примечание	Выполнение процедуры не приводит к остановке программы. Процедура игнорируется в случае если окно не было открыто

Процедура **dialog**

Функция	Определяет на экране диалоговое окно
Описание	*dialog (x1, y1, x2, y2)*
Пример	dialog(100,0,400,180) color(15) locate(10,20) name$=" " print "Введите имя файла" insertcontrol(100,20,180,36, name$)

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	N909 if inputdialog=0 then goto 909 closedialog if inputdialog<>1 then end  *Пример диалогового окна.*
Примечание	Параметры *x1, y1* представляют собой координаты верхнего левого угла диалогового окна, параметры *x2, y2* - это координаты правого нижнего угла. Правый левый угол экрана соответствует координате (1,1). Максимальный размер окна определяется как (1,1,800,540). Все координаты, указанные в процедурах (кроме самих координат окна) являются относительными координатами данного окна. Например, *locate(1,1)* всегда позиционирует курсор на верхний левый угол текущего окна. От текущего окна зависят все процедуры и функции вывода. На диалоговом окне расположены две кнопки - «OK» и «Отменить». При появлении диалогового окна на экране создается переменная *inputdialog,* которая доступна в программе, вызвавшей процедуру *dialog.* Начальное значение переменной *inputdialog* равно нулю. Если диалоговое окно было закрыто кнопкой «Отменить» или клавишей ESC, то значение переменой *inputdialog* становится равным -1. Если в диалоговом окне была нажата кнопка «OK» или клавиша ENTER, то значение переменой *inputdialog* равно +1. При помощи процедуры *insertcon-trol* в диалоговые окна можно вставлять строки ввода данных. Кроме того для диалоговых окон допустимы все операции вывода графической и текстовой информации.

Процедура **insertcontol**

Функция	Вставляет в диалоговое окно строку ввода данных
Описание	*insertcontrol (x1, y1, x2, y2 ,var)*
Пример	insertcontrol(20,40,120,60,alfa) insertcontrol(20,80,120,100,name$)
Примечание	Применяя процедуру *insertcontrol* можно вставить в диалоговое окно строку ввода данных. Координаты *x1,y1,x2,y2* определить положение строки в диалоговом окне. Переменная *var* может быть числовой или строковой. Не допускается использовать переменные другого типа. Значение переменной изменяется только когда нажата клавиша «OK» в диалоговом окне.

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Процедура **closedialog**

Функция	Закрывает диалоговое окно.
Описание	*closedialog*
Пример	dialog(100,0,400,180) color(15) locate(10,20) name$=" " print"Введите имя файла " insertcontrol(100,20,180,36,name$) N909 if inputdialog=0 then goto 909 Closedialog if inputdialog<>1 then end
Примечание	Выполнение процедуры не приводит к остановке программы. Процедура игнорируется если диалоговое окно не было открыто.

Системные функции

Функция **getdatacadr**

Функция	Возвращает данные управляющей программы, включая кадр, предшествующий функции.
Описание	*getdatacadr (n)*
Пример	Соответствие между номерами параметров и видом данных: 1..12- абсолютные координаты осей в текущей системе, заданные в управляющей программе. Порядок осей определяется в соответствии с базовыми станочными параметрами; 13..112 действующие на текущий момент G-функции: 13 - G0-G3; 17 - G4; 22 - G9; 23 - G10; 27 - G14,G15; 30 - G17-G20; 32 - G21,G22; 34 - G23,G24; 40 - G27; 41 - G28; 43 - G30,G31; 45 - G32; 46 - G33; 48 - G35,G36; 50 - G37; 51 - G38; 53 - G40-G42; 56 - G43,G44,G49; 58 - G45,G46; 66 - G53-G59,G92; 73 - G60;

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	74 - G50,G61,G62,G63; 77 - G64; 78 - G65,G66; 80 - G67,G68; 82 - G69; 93 - G80-G89; 103 - G90,G91; 107- G94, G95 109- G96, G97 113 - номер 1-й оси для плоскости, заданной G20; 114 - номер 2-й оси для плоскости, заданной G20; 115..126- текущие смещения нулей координат. Порядок осей определяется в соответствии с базовыми станочными параметрами; 127 - номер предыдущего кадра; 128 - номер корректора на радиус D; 129 - значение корректора на радиус; 130 - номер корректора на длину H; 131 - значение корректора на длину; 132- подача F; 133- частота вращения шпинделя S; 134- номер инструмента T; 135- время задержки E(G4); 136- текущее состояние шпинделя (3,4,5); 137-148 координаты точки касания в станочной системе координат 149-160 коэффициенты масштабирования по осям
Примечание	Параметр *n* это выражение целого типа, которое определяет вид возвращаемых

Функция **getsystemdata**

Функция	Возвращает данные системных переменных
Описание	*getsystemdata (n)*
Соответствие между номером параметра и видом данных	1..12 - заданное положение для оси координат с номером "n" - в микронах; 13..24 - абсолютное положение оси координат с номером "n-12"- в микронах 25..36 - заданное приращение(скорость) для оси координат с номером "n-24" - в микронах за такт таймера; 37..48 - код на ЦАП для оси координат с номером "n-36" без учета дрейфа нуля; 49..60 - значение положительного программного конечного выключателя для оси координат с номером "n-48" - в микронах; 61..72 - значение отрицательного программного конечного выключателя для оси координат с номером "n-60" - в микронах; 73..84 - значение параметров N7006, N7106 и т.д. (радиусное/диаметральное задание и индикация перемещений) 85 - значение параметра N3030 (действие диаметрального задания на IJK) 86 - значение параметра N2021 (способ задания IJK)

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	90 - номер отрабатываемого кадра 101..112 - значения добротностей по координатам 121 ..132 - значения коэффициентов скоростной компенсации по координатам 133..144 - заданное положение для осей координат в микронах с учетом перемещений в ручном режиме при установленной нечувствительности к ручным движениям 301..312 - значение аддитивного смещения нуля для функций G54-G59 для оси координат с номером "n-300" - в миллиметрах; 401..412 - значение смещения нуля для функции G54 для оси координат с номером "n-400" - в миллиметрах; 501..512 - значение смещения нуля для функции G55 для оси координат с номером "n-500" - в миллиметрах; 601..612 - значение смещения нуля для функции G56 для оси координат с номером "n-600" - в миллиметрах; 701..712 - значение смещения нуля для функции G57 для оси координат с номером "n-700" - в миллиметрах; 801..812 - значение смещения нуля для функции G58 для оси координат с номером "n-800" - в миллиметрах; 901..912 - значение смещения нуля для функции G59 для оси координат с номером "n-900" - в миллиметрах; 1001 ..1255 - значение корректора с номером "n-1000" - в миллиметрах; 1301..1316 - значение целочисленного параметра пользователя 2 группы с номером "n-1300"; 1401..1416 - значение вещественного параметра пользователя 1 группы с номером "n-1400"; 1500 - значение параметра N3033(способ задания тригонометрических функций); 1501 - значение параметра N2005(включена/выключена векторная коррекция); 1502 - значение параметра N1000(количество управляемых осей).
Примечание	Параметр *n* это выражение целого типа, которое определяет вид возвращаемых данных.

Процедура **setsystemdata**

ВАЖНО! Некорректное использование процедуры setsystemdata может привести к сбоям в работе системы, вызвать поломки станка.

Функция	Устанавливает новые значения системных переменных.
Описание	*setsystemdata (n, value)*
Соответствие между номером параметра и видом данных	1..12 - заданное положение для оси координат с номером "n" - в микронах; 13..24 - абсолютное положение оси координат с номером "n-12"- в микронах; 25..36 - заданное приращение(скорость) для оси координат с номером "n-24" - в микронах за такт таймера; 37..48 - код на ЦАП для оси координат с номером "n-36" без учета дрейфа

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	нуля; 49..60 - значение положительного программного конечного выключателя для оси координат с номером "n-48" - в микро¬нах; 61..72 - значение отрицательного программного конечного выключателя для оси координат с номером "n-60" - в микронах; 101..112 - значения добротностей по координатам 121..132 - значения коэффициентов скоростной компенсации по координатам 301..312 - значение аддитивного смещения нуля для функций G54-G59 для оси координат с номером "n-300" - в миллиметрах; 401..412 - значение смещения нуля для функции G54 для оси координат с номером "n-400" - в миллиметрах; 501..512 - значение смещения нуля для функции G55 для оси координат с номером "n-500" - в миллиметрах; 601..612 - значение смещения нуля для функции G56 для оси координат с номером "n-600" - в миллиметрах; 701..712 - значение смещения нуля для функции G57 для оси координат с номером "n-700" - в миллиметрах; 801 ..812 - значение смещения нуля для функции G58 для оси координат с номером "n-800" - в миллиметрах; 901 ..912 - значение смещения нуля для функции G59 для оси координат с номером "n-900" - в миллиметрах; 1001 ..1255 - значение корректора с номером "n-1000" - в миллиметрах;
Примечание	Параметр *n* это выражение целого типа и определяет номер системной переменной. Параметр *value* представляет собой выражение вещественного типа, содержащее новое значение системной переменной.

Процедура **geteadata**

Функция	Возвращает данные переменных модуля электроавтоматики.
Описание	*geteadata (group, index, var)*
Пример	window (20, 20, 200, 200) rem Чтение входного сигнала I2 geteadata(1,2, inp) rem Проверка наличия 3 бита во входном сигнале if (inp and 4)<>0 then goto 10 print "Бит 3 не установлен" goto 20 N10 print "Бит 3 установлен" N20 end
Соответствие между номером группы и видом данных	-Входы (байт) -Выходы (байт) -Динамическая память (байт) -Таймеры (текущее значение) (слово) -Счетчики (текущее значение) (слово) -Статическая память (байт) -Обменные ячейки (байт)
Примечание	Параметр *group* это выражение целого типа, определяющее группу

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	переменных, параметр *index* определяет адрес байта(слова) в группе. Информация из байта(слова) записывается в переменную *var.* Если указаны неверная группа или адрес байта в группе в переменную записывается ноль, программа продолжает свою работу и никакого сообщения не выдается.

Процедура **seteadata**

ВАЖНО! Некорректное использование процедуры seteadata может привести к сбоям в работе системы, вызвать поломки станка.

Функция	Устанавливает новые значения системных переменных.
Описание	*seteadata (group, index, value)*
Пример	*rem Установка динамической памяти M10* *mem = 48* *seteadata(3,10, mem)* *end*
Соответствие между номером группы и видом данных	-Входы (байт) -Выходы (байт) -Динамическая память (байт) -Таймеры - не обрабатываются -Счетчики (текущее значение) (слово) -Статическая память (байт) -Обменные ячейки (байт)
Примечание	Параметр *group* это выражение целого типа, определяющее группу переменных, параметр *index* определяет адрес байта(слова) в группе. Значение определяемое выражением *value* в указанный байт(слово). Процедура *seteadata* игнорируется, программа продолжает свою работу и никакого сообщения не выдается если указаны неверная группа или адрес байта в группе.

Процедура **gettooldata**

Функция	Возвращает данные из таблицы инструментов для заданного инструмента.
Описание	*gettooldata (num, index, var)*
Пример	rem Запрос смещения по оси 2 для инструмента 10 с размещением в переменной sm gettooldata(10,11,sm)
Соответствие между номером группы и видом данных	-длина инструмента -радиус инструмента -номер гнезда инструмента -тип инструмента -комментарий (строковый тип) 10..19 - смещения по осям 20..29 - позиция смены инструмента по осям 30..39 - аддитивные смещения по осям

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Примечание	Параметр *это* выражение целого типа и определяет номер инструмента, параметр *index* определяет вид запрашиваемых данных. Информация записывается в переменную *var.* Если указаны недопустимый номер инструмента, недопустимый вид данных или адрес переменной, выдается соответствующее сообщение.

Процедура **settooldata**

Функция	устанавливает данные в таблицу инструментов для заданного инструмента.
Описание	*settooldata (num, index, var)*
Пример	rem Установка длины для инструмента 10 из переменной abc settooldata(10,1 ,abc)
Соответствие между номером группы и видом данных	-длина инструмента -радиус инструмента -номер гнезда инструмента -тип инструмента -комментарий (строковый тип) 10..19 - смещения по осям 20..29 - позиция смены инструмента по осям 30..39 - аддитивные смещения по осям
Примечание	Параметр *num* это выражение целого типа и определяет номер инструмента, параметр *index* определяет вид устанавливаемых данных. Информация устанавливается из переменной *var.* Если указаны недопустимый номер инструмента, недопустимый вид данных или адрес переменной, выдается соответствующее сообщение.

Функция **paramactive**

Функция	Проверяет наличие параметра при вызове подпрограммы.
Описание	*paramactive (param)*
Пример	N10 P200 X12.34 Y45 H44 K73 ……. ……. :200 rem Проверка параметра X if paramactive (88) then goto 10 msg "Не задан параметр X" end N10 x=getparameter (88) rem Проверка параметра Y if paramactive (89) then goto 20 msg "Не задан параметр Y" end N20 y=getparameter (89) rem Проверка параметра H if paramactive (72) then goto 30

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	msg "Не задан параметр H" end N30 h=getparameter (72) rem Проверка параметра K if paramactive (75) then goto 40 msg "Не задан параметр K" end N40 k=getparameter (75) M99
Соответствие между номером группы и видом данных	-длина инструмента -радиус инструмента -номер гнезда инструмента -тип инструмента -комментарий (строковый тип) 10..19 - смещения по осям 20..29 - позиция смены инструмента по осям 30..39 - аддитивные смещения по осям
Примечание	Параметр *param* это выражение целого типа в диапазоне от 65 до 90 и определяет, был ли указан параметр при вызове подпрограммы *P.* Числовое значение параметра соответствует коду ASCII символов от "A" до "Z" латинского алфавита. если параметр не был указан и равно 1, если параметр был указан, то значение функции равно 0,. Появляется сообщение об ошибке и программа останавливается если значение параметра *param* выходит за пределы диапазона 65..90. Функцию paramactive можно использовать только при вызове подпрограммы с помощью символа P. Эта функция бессмысленна при вызове подпрограмм с помощью процедуры gosub

Функция **getparameter**

Функция	Возвращает значение соответствующего параметра, заданного в подпрограмме
Описание	*getparameter (param)*
Пример	N10 P200 X12.34 Y45 H44 K73 …… …… :200 rem Проверка параметра X if paramactive (88) then goto 10 msg “Не задан параметр X” end N10 x=getparameter (88) rem Проверка параметра Y if paramactive (89) then goto 20 msg “Не задан параметр Y” end N20 y=getparameter (89) rem Проверка параметра H

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	if paramactive (72) then goto 30 msg “Не задан параметр H” end N30 h=getparameter (72) rem Проверка параметра K if paramactive (75) then goto 40 msg “Не задан параметр K” end N40 k=getparameter (75) …. M99

Символ	*Код*
A	65
B	66
C	67
D	68
E	69
F	70
G	71
H	72
I	73
J	74
K	75
L	76
M	77
N	78
O	79
P	80
Q	81
R	82
S	83
T	84
U	85
V	86
W	87
X	88
Y	89
Z	90

Примечание	Параметр *param* это выражение целого типа в диапазоне от 65 до 90 и определяет, был ли указан параметр при вызове подпрограммы *P.* Числовое значение параметра соответствует коду ASCII символов от "A" до "Z" латинского алфавита. если параметр не был указан и равно 1, если параметр был указан, то значение функции равно 0,. Появляется сообщение об ошибке и программа останавливается если значение параметра *param* выходит за пределы диапазона 65..90. Функцию paramactive можно использовать только при вызове подпрограммы с помощью символа P. Эта функция бессмысленна при вызове подпрограмм с помощью процедуры gosub

Процедура **dump**

Функция	Выводит на экран значения всех переменных, которые используются в программе.
Описание	*dump*
Примечание	Процедура *dump* открывает на экране окно и выводит в него информацию

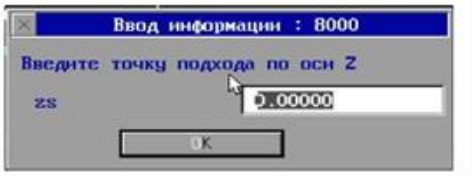
ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	о значениях всех переменных, используемых в программе. Эта процедура используется для отладки.

Процедура **rem**

Функция	Весь дальнейший текст (до конца строки) считается комментарием.
Описание	*rem*
Примечание	При помощи процедуры *rem можно* вводить в текст комментарии, поясняющие работу программы. В процессе отладки можно временно "закомментировать" определенные операторы.

Процедура **input**

Функция	Вводит с клавиатуры числовых или строковых данных в переменную.
Описание	*input "comment", inp*
Пример	**input "Введите точку подхода по оси Z", zs** *Диалоговое окно ввода информации* 
Примечание	С помощью процедуры *input в процессе работы программы* можно вводить значения числовых или строковых переменных. При вызове процедуры input на экране появляется диалоговое окно ввода информации. Параметр *"comment"* является комментарием, который выводится в окне. Этот параметр можно опускать. Параметр inp является переменной, в которую заносится информация из диалогового окна.

Файловые операции ввода-вывода

Процедура **open**

Функция	Открывает файл на диске для чтения, записи или добавления.
Описание	*open filename for mode as #number*
Пример	**open "c:\report.txt" for append as #4** **open "d:\data.tap" for input as #2** **open "c:\program\out.dat" for output as #5**
Примечание	Процедура *open* открывает файл для ввода, вывода или добавления. Параметр *filename* считается выражением(переменной) строкового типа и представляет собой имя файла. Соглашения по допустимым именам файлов такие же, как в операционной системе MS-DOS, то есть длина имени 8 символов, а расширения - 3 символа. Символы -;, = + <>|"[]\,и символ пробела не допускается использовать в именах. Параметр *mode* может принимать 3 значения input (открыть существующий файл для ввода из него информации); output (создать, а если существует, то заменить файл для

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

	записи в него информации); append (открыть существующий файл для записи в него информации, она записывается в конец файла, добавляется к уже существующей). Параметр number является идентификатором файла для операций read и write и принимает значения от 1 до 9, то есть можно одновременно открыть не более 9 файлов.

Процедура **real**

Функция	Считывает одно или более значений из файла в одну или более переменных.
Описание	*read #number a, b, c ,...*
Примечание	Параметр *number* считается идентификатором файла (см. процедуру *open*). Каждый параметр *a, b, c, …* является переменной строкового или числового типа. При переменной числового типа процедура *read* считывает из файла последовательность символов, которые образуют число со знаком. Пропускаются пробелы, знаки табуляции или метки конца строки, которые предшествуют числовой строке. Считывание заканчивается при обнаружении первого пробела, символа табуляции, не цифры или метки конца строки, следующих за числовой строкой, или в том случае, если файл заканчивается. Происходит ошибка и программа останавливается при не соответствии числовая строка ожидаемому формату. В ином случае переменной присваивается значение. Переменной присваивается нулевое значение если файл закончился до появления числа. Последующая операция *read* начнется с пробела, символа табуляции либо метки конца строки, которыми закончилась числовая строка. В случае переменной строкового типа процедура *read* считывает все без исключения символы, до следующей метки конца строки (но исключая ее), или пока файл не закончится, или пока размер результирующей строки не дойдет до 255 символов. Переменной присваивается получившаяся в результате символьная строка. Следующая операция *read* начинается с метки конца строки, которой закончилась предыдущая строка.
Ограничение	Процедура*read* работает только с файлами открытыми для ввода информации.

Процедура **write**

Функция	Записывает значение одной или более переменных в файл
Описание	*write #number a, b, c ,...*
Примечание	Параметр *number* считается идентификатором файла (см. процедуру *open*). Если использовать параметр *number* равным нулю, то значения переменных записываются в окно текстового редактора при условии, что оно открыто.
Ограничение	Процедура *write* работает только с открытыми для вывода или добавления (output или append) информации файлами.

Процедура **close**

Функция	Закрывает файл.

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Описание	*close #number*
Примечание	Параметр *number* считается идентификатором файла (см. процедуру *open).*

Функция **eof**

Функция	Возвращает состояние файла
Описание	*eof(number)*
Примечание	Параметр *number* считается идентификатором файла (см. процедуру *open*). Значение функции равно 1, если внутренний указатель позиции находится в конце файла и равно 0 в противном случае., Появляется сообщение об ошибке и программа останавливается в случае если файл не открыт., Появляется сообщение об ошибке и программа останавливается в случае если значение параметра number больше 9 или меньше 1.

Перечень сообщений об ошибках в макропрограммах

Арифметическое переполнение	при арифметическом действии результат становится больше максимально возможного значения для параметра, выражения или переменной использующейся для хранения результата
Деление на ноль	
Длина строки более 255 символов	длина результирующей строковой переменной после
Должен быть оператор "as"	не задан оператор *as* в процедуре *open*
Должно быть число, переменная или функция	в выражении недопустимое сочетание символов
Корень из отрицательного значения	
Недопустимое значение	недопустимое значение параметра, переменной,выражения
Недопустимое значение аргумента функции	значение аргумента функций *tan, lg, ln* находится вне области определения этих функций
Недопустимое имя файла	не соответствующее требованиям DOS имя файла в процедуре open
Недопустимый номер	в файловых процедурах, номера параметра в процедурах и функциях *getdatacadr, getsystemdata, setsystemdata, geteadata, seteadata* недопустимое значение индекса массива, идентификатора файла
Не найден оператор "else"	отсутствует оператор *else* в конструкции *iff*
Не найдена метка	не найдена точка перехода при выполнении операторов *goto* или *gosub*
Нет знака =	

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru

Нет оператора "endiff"	отсутствует оператор *endiff* в конструкции *iff*
Нет оператора "for"	
Нет оператора "next"	
Нет оператора "then"	
Нет оператора "to"	
Нет открытого оператора "iff"	задание оператора *else* без *iff;*
Нет"("	
Нет")"	
Не указан идентификатор файла	
Не указан режим открытия файла (input,output,append)	
Ожидается конец строки или кадра	заданы другие выражения, процедуры и функции в строке с процедурами работы с файлами дополнительно;
Ожидается переменная	вместо переменной использование константы
Ожидается число	вместо константы использование переменной
Ожидается ","	
Ожидается "	
Оператор"return"без оператора "gosub"'	
Отсутствует оператор	символы = <> пропущены
Ошибка доступа к файлу	
Ошибка чтения числа из файла	встретилось не число при чтении в файле
Синтаксическая ошибка	символ или сочетание символов недопустимы
Требуется строковая переменная	пересылка строковых данных в переменную другого типа
Файл не открыт	
Файл не открыт для ввода	
Файл с данным идентификатором уже открыт	задание процедуры *open* с уже используемым идентификатором файла #
Функции *getstring* или *restore* используются не в подпрограмме	

ООО «Новые Электронные Технологии»
Ростовская область, г. Азов, ул. Промышленная, д.2.
Телефон (863)298-22-99
www.chpu.net elektronika-net@mail.ru