

УТВЕРЖДЕН
ГУКН.421457.001 03 34 6206-1-ЛУ

КОМПЛЕКС ПРОГРАММНЫХ СРЕДСТВ
«КАСКАД-САУ»

Редактор алгоритмов ИЕС 1131-3

Руководство оператора

Приложение 4
Поддерживаемые конструкции языка Structured Text

ГУКН.421457.001 03 34 6206-5
Листов 26

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

1 Вставка кода	2
2 Конструкции языка	3
2.1 Приоритеты операций	4
2.2 Арифметические операции	4
2.3 Логические (побитовые) операции.....	4
2.4 Операции сравнения	4
2.5 Присвоение	5
2.6 Конструкция IF – ELSEIF – ELSE	5
2.7 Цикл FOR.....	5
2.8 Цикл WHILE.....	7
2.9 Цикл REPEAT UNTIL.....	7
2.10 Конструкция CASE	7
3 Вызов функций и блоков.....	10
3.1 Функции приведения типов	10
3.2 Тригонометрические и алгебраические функции	10
3.3 Возведение в степень	11
3.4 Битовые сдвиги	11
3.5 Функции выбора.....	11
3.5.1 Выбор минимума и максимума	11
3.5.2 Выбор по ключу	11
3.6 Управление TIME-переменной.....	12
3.7 Работа с датой и временем	12
3.8 Доступ к полям точки контроллера «Каскад-САУ».....	15
3.9 Системные возможности контроллера «Каскад-САУ».....	15
3.10 Газовые расчеты.....	17
3.11 Работа с дискретным сигналом.....	24
3.12 Вызов блоков и схем, разработанных пользователем	25

1 ВСТАВКА КОДА

Для вставки кода на структурированном текстовом языке (в дальнейшем будет использоваться русское сокращение «СТ») необходимо:

- 1) Добавить на поле редактирования элемент «Вычисляемый текст» (находится на странице «Элементы» окна с набором стандартных элементов);
- 2) Дважды щелкнуть левой клавиши мыши на появившемся элементе или выбрать в контекстном меню пункт ТЕКСТ.

При написании кода на СТ допустимо использовать конструкции языка, стандартные функции и блоки, а также вызовы блоков и схем, описанных пользователем; программа, написанная на СТ должна быть ограничена парой служебных слов BEGIN ... END; (символ «;» после END обязателен).

2 КОНСТРУКЦИИ ЯЗЫКА

К конструкциям языка СТ относятся:

- арифметические операции
- логические (побитовые) операции
- операции сравнения
- операция присвоения
- конструкция IF – ELSEIF – ELSE
- цикл FOR
- цикл WHILE
- цикл REPEAT UNTIL
- конструкция CASE

При записи арифметических выражений допустимо использование скобок для указания порядка вычислений.

При записи выражений допустимо использовать переменные (локальные и глобальные) и константы. Таблица 1 содержит список типов констант и соответствующих им форматов записи.

Т а б л и ц а 4.1

Тип	Формат записи
FLOAT	Для указания типа FLOAT необходимо использовать десятичную точку (константа без десятичной точки принимается за целочисленную). Например: 12.34 – FLOAT константа, но 12 – целая константа
INTEGER	Для записи целых констант допустимо использовать только цифры. Например: 123 – целая константа
TIME	Для указания значения времени необходимо использовать следующий вид записи [nD[nH[nM]]]nSn, где D – число дней, H – число часов M – число минут S – число секунд После знака «S» указывается число миллисекунд. Например: 12s10 – означает 12 секунд и 10 миллисекунд 1h10s0 – 1 час, 10 секунд
BOOLEAN	Для задания значений типа BOOLEAN используются ключевые слова TRUE и FALSE
DATETIME	Для указания даты и времени необходимо использовать следующий вид записи: ‘YYYY-MM-DD-HH:MnMn:SS’, где YYYY – год MM – месяц DD – день

	HH – часы MnMn – минуты SS – секунды. Значение даты и времени ДОЛЖНО БЫТЬ заключено в апострофы. Например: Var := '2009-09-01-09:00:00' – переменной Var присваивается момент начала торжественной линейки 1 сентября 2009 года.
--	---

2.1 Приоритеты операций

Приоритет операций в выражениях указан в таблице 2 (чем выше приоритет, тем раньше выполняется операция):

Т а б л и ц а 4.2

Операция	Приоритет
Сравнения	1
Сложение, вычитание	2
Умножение, деление	3
OR	4
AND, XOR	5
NOT	6
Унарный минус	7
Вызов функции	8

2.2 Арифметические операции

К арифметическим операциям относятся:

- «+» - сложение
- «-» - вычитание
- «*» - умножение
- «/» - деление
- «mod» - остаток от целочисленного деления

2.3 Логические (побитовые) операции

К данным операциям относятся:

- «OR» - Логическое (побитовое) сложение
- «AND» - Логическое (побитовое) умножение
- «XOR» - Логическое (побитовое) исключающее ИЛИ
- «NOT» - Логическое (побитовое) отрицание

2.4 Операции сравнения

Поддерживаются следующие операции сравнения:

- «=» - сравнение на равенство
- «<>» - сравнение на неравенство

- «>» - сравнение на больше
- «>=» - сравнение на не меньше
- «<» - сравнение на меньше
- «<=» - сравнение на не больше

В качестве результата сравнения всегда используется значение типа BOOLEAN.

2.5 Присвоение

Для обозначения присвоения используется парный знак «:=». В правой и левой части выражения должны быть операнды одного типа (автоматического приведения типов не предусмотрено). В левой части выражения (принимающая сторона) может быть использована только переменная. Правая часть может содержать выражение или константу.

2.6 Конструкция IF – ELSEIF – ELSE

Здесь и далее по тексту считается, что:

- выражение в фигурных скобках может использоваться ноль или больше раз подряд
- выражение в квадратных скобках не обязательно к использованию

Конструкция IF-ELSEIF-ELSE имеет следующий формат:

```
IF <boolean expression> THEN <statement list>  
[ELSEIF <boolean expression> THEN <statement list>]  
[ELSE <statement list>]  
END_IF;
```

Например:

```
if Var <> 0  
  then Var := 1  
  elseif Var > 0  
    then Var := 0;  
  else Var := 10;  
end_if;
```

Конструкция допускает вложенность, т.е. внутри одного IF может быть еще один и т.д.
Например:

```
if Var > 10 then  
  if Var < Var2 + 1  
    then Var := 10;  
    else Var := 0;  
  end_if;  
end_If;
```

2.7 Цикл FOR

Служит для задания цикла с фиксированным количеством итераций. Формат конструкции следующий:

```

FOR <Control Variable> := <expression1> TO <expression2>
[BY <expression3>] DO
    <statement list>
END_FOR;

```

При задании условий цикла считается, что ControlVariable, Expression1 ... Expression3 имеют тип INTEGER. Выход из цикла будет произведен в том случае, если значение переменной цикла превысит значение Expression2. Например:

```

For i := 1 to 10 by 2 do
    k := k * 2;
End_for;

```

Оператор BY задает приращение переменной цикла (в данном случае I будет увеличиваться на 2 при каждом проходе по циклу). Если оператор BY не указан, то приращение равно 1. Например:

```

For i := 1 to k / 2 do
    Var := Var + k;
    k := k - 1;
End_for;

```

Внутри цикла могут использоваться другие циклы, операторы IF и CASE. Для выхода из цикла (любого типа) может использоваться оператор EXIT. Например

```

For i := 1 to 10 by 2 do
    k := k * 2;
    if k > 20 then
        Exit;
    end_if;
End_for;

```

Примечание 1: Выражения Expression1 ... Expression3 вычисляются до входа в цикл, поэтому изменения значений переменных, входящих в любое из этих выражений не приведет к изменению числа итераций. Например:

```

01: k := 10;
02: for I := 1 to k / 2 do
03:     k := 20;
04: end_for;

```

В строке 3 производится изменение переменной k, но цикл все равно выполнится только пять раз.

Примечание 2: Значение переменной цикла может изменяться внутри тела цикла, но в начале очередной итерации значение данной переменной будет выставлено в соответствие с условиями цикла. Например:

```

01: for I := 1 to 5 do
02:     I := 55;
03: end_for;

```

При первом проходе значение I будет равно 1, потом в строке 2 изменится на 55, но на втором проходе значение I станет равно 2 – следующему значению по условиям цикла.

2.8 Цикл WHILE

Служит для определения цикла с предусловием. Цикл будет исполняться до тех пор, пока выражение в предложении WHILE возвращает TRUE. Формат конструкции следующий:

```
WHILE <Boolean-Expression> DO  
  <Statement List>  
END_WHILE;
```

Значение <Boolean-Expression> проверяется на каждой итерации. Завершение цикла произойдет, если выражение <Boolean-Expression> вернет FALSE. Например:

```
k := 10;  
While k > 0 do  
  i := I + k;  
  k := k - 1;  
End_while;
```

Внутри цикла могут использоваться другие циклы, операторы IF и CASE. Для досрочного завершения цикла используется оператор EXIT (см. пример в описании цикла FOR).

2.9 Цикл REPEAT UNTIL

Служит для определения цикла с постусловием. Завершение цикла произойдет тогда, когда выражение в предложении UNTIL вернет FALSE. Другими словами: цикл будет выполняться, пока условие в предложении UNTIL не выполнится. Формат конструкции следующий:

```
REPEAT  
  <Statement List>  
UNTIL <Boolean Expression>;  
END_REPEAT;
```

Например:

```
k := 10;  
REPEAT  
  i := i + k;  
  k := k - 1;  
UNTIL k = 0;  
END_REPEAT;
```

Внутри цикла могут использоваться другие циклы, операторы IF и CASE. Для досрочного завершения цикла используется оператор EXIT (см. пример в описании цикла FOR).

2.10 Конструкция CASE

Данная конструкция служит для организации выбора из диапазона значений. Формат конструкции следующий:

```
CASE <Expression> OF  
  CASE_ELEMENT {CASE_ELEMENT}  
  [ELSE <Statement List>]
```

END_CASE;

CASE_ELEMENT – это список значений, перечисленных через запятую. Элементом списка может быть целое число или диапазон целых чисел. Диапазон задается следующим образом BEGIN_VAL .. END_VAL.

Если текущее значение Expression не попало ни в один CASE_ELEMENT, то управление будет передано на предложение ELSE. Если предложение ELSE не указано, то никаких действий выполнено не будет.

Значение Expression может быть только целым. Например:

```
01: CASE k OF
02:   1:
03:     k := k * 10;
04:   2..5:
05:     k := k * 5;
06:     i := 0;
07:   6, 9..20:
08:     k := k - 1;
09:   ELSE
10:     k := 0;
11:     i := 1;
12: END_CASE;
```

Строка 4 содержит диапазон значений. Если значение K принадлежит числовому отрезку [2, 5], то будут выполнены строки 5 и 6.

В строке 7 использован список значений. Строка 8 выполнится, если значение K будет равно 6 или будет принадлежать числовому отрезку [9, 20].

Строки 10 и 11 будут выполнены в том случае, если $K < 1$, или $6 < K < 9$, или $K > 20$ (в данном случае сработает предложение ELSE).

При задании списка значений необходимо выполнять следующие условия:

- Наборы значений внутри одного CASE не должны пересекаться
- При указании диапазона значений начало диапазона должно быть меньше его конца.

Таблица 3 содержит образцы правильной и неправильной записи конструкции CASE.

Т а б л и ц а 3

Неправильная запись	Правильная запись
<pre>01: CASE k OF 02: 1: 03: k := k * 10; 04: 2..5: 05: k := k * 5; 06: i := 0; 07: 5, 9..20: 08: k := k - 1; 09: ELSE 10: k := 0; 11: i := 1; 12: END_CASE;</pre> Диапазоны в строках 4 и 7 пересекаются	<pre>01: CASE k OF 02: 1: 03: k := k * 10; 04: 2..5: 05: k := k * 5; 06: i := 0; 07: 6, 9..20: 08: k := k - 1; 09: ELSE 10: k := 0; 11: i := 1; 12: END_CASE;</pre>
<pre>01: CASE k OF 02: 1: 03: k := k * 10; 04: 2..5: 05: k := k * 5; 06: i := 0; 07: 6, 20..9:</pre>	<pre>01: CASE k OF 02: 1: 03: k := k * 10; 04: 2..5: 05: k := k * 5; 06: i := 0; 07: 6, 9..20:</pre>

<pre>08: k := k - 1; 09: ELSE 10: k := 0; 11: i := 1; 12: END_CASE;</pre> В строке 7 диапазон значений задан неправильно.	<pre>08: k := k - 1; 09: ELSE 10: k := 0; 11: i := 1; 12: END_CASE;</pre>
--	---

Действия, предусмотренные для обработки каждого из случаев CASE, могут использовать циклы, операторы IF и CASE.

3 ВЫЗОВ ФУНКЦИЙ И БЛОКОВ

При написании программ на СТ возможно использование стандартных функций, специализированных функций и блоков «Каскад-САУ», а так же блоков и схем, разработанных пользователем.

3.1 Функции приведения типов

Приведение FLOAT к указанному типу (путем округления):

```
F_I (PARAM: FLOAT) : INTEGER  
F_B (PARAM: FLOAT) : BOOLEAN  
F_Tm (PARAM: FLOAT) : TIME  
F_DT (PARAM: FLOAT) : DATETIME
```

Приведение FLOAT к указанному типу (путем отсечения дробной части):

```
TF_I (PARAM: FLOAT) : INTEGER  
TF_B (PARAM: FLOAT) : BOOLEAN  
TF_Tm (PARAM: FLOAT) : TIME
```

Приведение INTEGER к указанному типу:

```
I_F (PARAM: INTEGER) : FLOAT  
I_Tm (PARAM: INTEGER) : TIME  
I_B (PARAM: INTEGER) : BOOLEAN
```

Приведение TIME к указанному типу:

```
Tm_F (PARAM: TIME) : FLOAT  
Tm_B (PARAM: TIME) : BOOLEAN
```

Приведение BOOLEAN к указанному типу:

```
B_F (PARAM: BOOLEAN) : FLOAT  
B_I (PARAM: BOOLEAN) : INTEGER
```

Приведение DATETIME к указанному типу:

```
DT_F (PARAM: DATETIME) : FLOAT
```

3.2 Тригонометрические и алгебраические функции

Все перечисленные в данном разделе функции принимают аргументы типа FLOAT и возвращают результат типа FLOAT. Формат вызова следующий:

```
FUNCTION (PARAM: FLOAT) : FLOAT
```

где FUNCTION - одна из следующих функций:

- ABS – модуль
- SQRT – квадратный корень
- LN – натуральный логарифм
- LOG – десятичный логарифм

- EXP – экспонента
- SIN – синус
- COS – косинус
- TAN – тангенс
- ASIN – арксинус
- ACOS – арккосинус
- ATAN – арктангенс

3.3 Возведение в степень

Для возведения в степень используется функция EXPT, которая имеет следующий формат:

```
EXPT(RADIX: FLOAT, EXPONENT: FLOAT): FLOAT
```

где RADIX – основание степени, EXPONENT – показатель степени.

3.4 Битовые сдвиги

Все перечисленные в данном разделе функции принимают два аргумента типа INTEGER, из которых первый означает ЧТО СДВИГАТЬ, второй – НА СКОЛЬКО СДВИГАТЬ. Формат вызова следующий:

```
FUNCTION(PARAM1: INTEGER, PARAM2: INTEGER): INTEGER
```

где FUNCTION – одна из следующих функций:

- SHL – сдвиг влево
- SHR – сдвиг вправо
- ROL – циклический сдвиг влево
- ROR – циклический сдвиг вправо

3.5 Функции выбора

3.5.1 Выбор минимума и максимума

Функции выбора Min/Max принимают по два аргумента ОДИНАКОВОГО ТИПА и возвращают значение того же типа, что и аргументы. Формат вызова следующий:

```
FUNCTION(PARAM1: TYPE, PARAM2: TYPE): TYPE
```

TYPE может быть FLOAT, INTEGER, BOOLEAN, TIME, DATETIME

FUNCTION – одна из следующих функций:

- MAX – выбрать большее из двух
- MIN – выбрать меньшее из двух

3.5.2 Выбор по ключу

Функции выбора по ключу:

- SELECT
- LIMIT

– MUX

Функция SELECT имеет следующий формат:

```
SELECT(G: BOOLEAN, IN1: TYPE, IN2: TYPE): TYPE
```

где TYPE может быть FLOAT, INTEGER, BOOLEAN, TIME, DATETIME. Результат выполнения функции:

- Result = IN1, если G = FALSE
- Result = IN2, если G = TRUE

Функция LIMIT имеет следующий формат:

```
LIMIT(MIN: TYPE, IN: TYPE, MAX: TYPE): TYPE
```

где TYPE может быть FLOAT, INTEGER, BOOLEAN, TIME, DATETIME. Результат выполнения функции:

- Result = MIN, если IN < MIN
- Result = MAX, если IN > MAX
- Result = IN, если MIN <= IN <= MAX

Функция MUX имеет следующий формат:

```
MUX(K: INTEGER, IN1: TYPE, IN2: TYPE): TYPE
```

где TYPE может быть FLOAT, INTEGER, BOOLEAN, TIME, DATETIME. Результат выполнения функции:

- Result = IN1, если K = 0
- Result = IN2, если K = 1

3.6 Управление TIME-переменной

Запуск накопления времени имеет следующий формат:

```
TSTART(IN: TIME)
```

где IN может быть только TIME-переменной. Вызов ничего не возвращает.

Останов накопления времени имеет следующий формат:

```
TSTOP(IN: TIME)
```

где IN может быть только TIME-переменной. Вызов ничего не возвращает.

3.7 Работа с датой и временем

Для работы с датой и временем доступны следующие функции и блоки:

- «DT_Get» – получить дату и время в формате DATETIME;
- «DT_Pack» – упаковать дату и время из формата «ГОД.МЕСЯЦ.ДЕНЬ + ЧАС:МИНУТА:СЕКУНДА.МИЛЛИСЕКУНДА» в формат DATETIME. При упаковке локальное время будет переведено во время по Гринвичу;
- «DT_Unpack» – развернуть дату и время из формата DATETIME в локальное время в формате «ГОД.МЕСЯЦ.ДЕНЬ + ЧАС:МИНУТА:СЕКУНДА.МИЛЛИСЕКУНДА»;
- «DT_Diff» – получить разность времен (DT1 – DT2);

- «DT_Ink» – изменить дату и время (вход iDT) на указанное число лет, месяцев, дней, часов, минут, секунд и миллисекунд;
- «DT_Sched» – получить число отчетных моментов в указанном интервале (DT2 – DT1);
- «DT_Last_Sched» - получить время последнего наступления контрактного года, месяца, декады, дня, часа, начиная с указанного времени до текущего момента.

Получение даты и времени

DT_Get: DATETIME

Функция не имеет параметров.

Упаковка даты и времени

DT_Pack(YRS, MON, DAY, HRS, MIN, SEC, MS, DST, DT, ERR)

Входные параметры:

YRS, MON, DAY, HRS, MIN, SEC, MS: INTEGER,

Год, месяц, день, часы, минуты, секунды, миллисекунды соответственно.

DST: BOOLEAN.

Признак летнего/зимнего времени. Время летнее, если DST = TRUE.

Выходные параметры:

DT: DATETIME

Дата и время по Гринвичу в формате DATETIME.

ERR: BOOLEAN.

ERR равен TRUE, если в процессе упаковки произошла ошибка.

Распаковка даты и времени

DT_Unpack(DT, YRS, MON, DAY, HRS, MIN, SEC, MS, DST, ERR)

Входные параметры:

DT: DATETIME.

Дата и время по Гринвичу в формате DATETIME.

Выходные параметры:

YRS, MON, DAY, HRS, MIN, SEC, MS: INTEGER

Год, месяц, день, час, минута, секунда, миллисекунда в локальном времени APM.

DST: BOOLEAN

DST вернет TRUE, если время летнее.

ERR: BOOLEAN

ERR вернет TRUE, если в процессе распаковки произошла ошибка.

Получение разности времен

DT_Diff(DT1, DT2, Hrs, Min, Sec, mS)

Входные параметры:

DT1, DT2: DATETIME

Времена, между которыми определяется разница.

Выходные параметры:

HRS: INTEGER

Разница в полных часах.

MIN: INTEGER

Разница в полных минутах.

SEC: INTEGER

Разница в полных секундах.

MS: TIME

Разница в миллисекундах.

Изменение даты и времени

DT_Ink(iDT, Yrs, Mon, Day, Hrs, Min, Sec, mSec, oDT, Err)

Входные параметры:

iDT: DATETIME

Исходная дата и время.

YRS, MON, DAY, HRS, MIN, SEC, MSEC: INTEGER

Параметры определяют, на сколько лет, месяцев и т.д. изменить исходную дату и время. Приращение может быть как положительным, так и отрицательным.

Выходные параметры:

oDT: DATETIME

Новое значение даты и времени.

ERR: BOOLEAN

ERR вернет TRUE, если в процессе изменения даты и времени произошла ошибка.

Получение числа отчетных моментов в указанном интервале

DT_Sched(DT1, DT2, Hour, Min, DST, Yrs, Mon, Dec, Day, Hrs, Err)

Входные параметры:

DT1, DT2: DATETIME

Границы интервала, внутри которого необходимо получить число отчетных моментов.

HOUR, MIN: INTEGER

Задают контрактный час (с точностью до минуты).

DST: BOOLEAN

Если подано TRUE, то границы интервала заданы в летнем времени.

Выходные параметры:

YRS, MON, DEC, DAY, HRS: INTEGER

Параметры возвращают кол-во наступивших отчетных лет, месяцев, декад, дней и часов.

ERR: BOOLEAN

ERR вернет TRUE, если произошла ошибка вычислений.

Получение времени последнего наступления отчетного года, месяца, декады, дня, часа, начиная с указанного времени до текущего момента

DT_Last_Sched(iDT, iHrs, iMin, DST, LYrs, LMon, LDec, LDay, LHrs, Err)

Входные параметры:

iDT: DATETIME

Границы интервала, внутри которого необходимо получить время последнего наступления отчетного года, месяца, декады, часа.

iHRS, iMIN: INTEGER

Задают контрактный час (с точностью до минуты).

DST: BOOLEAN

Если подано TRUE, то границы интервала в летнем времени.

Выходные параметры:

LYRS, LMON, LDEC, LDAY, LHRS: DATETIME

Дата и время последнего наступления отчетного года, месяца, декады, часа.

ERR: BOOLEAN

Вернет TRUE, если произошла ошибка.

3.8 Доступ к полям точки контроллера «Каскад-САУ»

Функции получения значений полей точки принимают на вход номер точки (PT: INTEGER) и возвращают FLOAT-значение, кроме функции Get_Status, которая возвращает INTEGER. Формат вызова следующий:

```
FUNCTION (PT: INTEGER) : FLOAT или
Get_STATUS (PT: INTEGER) : INTEGER
```

Где FUNCTION может быть:

- Get_LLL
- Get_LL
- Get_L
- Get_H
- Get_HH
- Get_HHH
- Get_TE_MIN
- Get_TE_MAX
- Get_D_MIN
- Get_D_MAX
- Get_TRACT
- Get_VALUE

Блоки установки значения и статуса точки имеют следующий формат:

```
Put_Value (PT: INTEGER, VALUE: FLOAT)
Put_Status (PT: INTEGER, STATUS: INTEGER)
```

3.9 Системные возможности контроллера «Каскад-САУ»

Получить текущую не критическую ошибку интерпретатора

GET_ERROR: INTEGER

Нет параметров.

Сбросить текущую некритическую ошибку интерпретатора

CLEAR_ERROR

Нет параметров.

Установка режима технологического процесса

ProcessMode (ON: BOOLEAN)

Входной параметр ON указывает включить (TRUE) или выключить (FALSE) режим технологического процесса.

Записать значение точки в локальную БД контроллера «Каскад-САУ»

SavePointVal (Pt: INTEGER)

Параметр PT – номер точки.

Записать значение статуса точки в локальную БД контроллера «Каскад-САУ»

SavePointStatus (Pt: INTEGER)

Параметр PT – номер точки.

Получить из локальной БД контроллера «Каскад-САУ» значение и статус указанной точки

RestorePoint (Pt: INTEGER, Value: FLOAT, STATUS: INTEGER,
IsValue: BOOLEAN, IsStatus: BOOLEAN)

Входы:

PT: INTEGER

Номер точки.

Выходы:

VALUE: FLOAT

Значение точки. Имеет смысл, если ISVALUE равно TRUE.

STATUS: INTEGER

Статус точки. Имеет смысл, если ISSTATUS равно TRUE.

ISVALUE: BOOLEAN

Признак того, что удалось прочитать значение указанной точки.

ISSTATUS: BOOLEAN

Признак того, что удалось прочитать статус указанной точки.

3.10 Газовые расчеты

Расчет коэффициента сжимаемости газа согласно ГОСТ 30319.2-96 по методу NX19

GasK_NX19(IN1, IN2, ..., OUT1, OUT2, ...)

Входы:

Gas_Temp: FLOAT
температура газа [K]
Gas_Press: FLOAT
давление газа [МПа]
Gas_Density: FLOAT
плотность газа [кг/м³]
Molar_N2: FLOAT
молярная часть N₂
Molar_CO2: FLOAT
молярная часть CO₂

Выходы:

Error_Code: INTEGER
код ошибки: 0 - нет ошибки, 1 - неверные входные аргументы
Compress_Coeff : FLOAT
рассчитанный коэффициент сжимаемости газа

Расчет коэффициента сжимаемости газа согласно ГОСТ 30319.2-96 по методу GERG91

GasK_GERG91(IN1, IN2, ..., OUT1, OUT2, ...)

Входы:

Gas_Temp: FLOAT
температура газа [K]
Gas_Press: FLOAT
давление газа [МПа]
Gas_Density: FLOAT
плотность газа [кг/м³] при нормальных условиях (p = 1atm, t = 20C)
Molar_N2: FLOAT
молярная часть N₂
Molar_CO2: FLOAT
молярная часть CO₂

Выходы:

Error_Code: INTEGER
код ошибки: 0 - нет ошибки, 1 - неверные входные аргументы
Compress_Coeff: FLOAT
рассчитанный коэффициент сжимаемости газа

Расчет расхода газа при прохождении через сужающее устройство с угловым отбором давления согласно ГОСТ 8.563.2-97

GasFlowrate_DA(IN1, IN2, ..., OUT1, OUT2, ...)

Входы:

Gas_Temp: FLOAT
температура газа [K]
Gas_Press: FLOAT
давление газа [МПа]
Gas_Density: FLOAT
плотность газа [кг/м³] при нормальных условиях (p=1atm, t=20C)
Molar_N2: FLOAT
молярная часть N2
Molar_CO2: FLOAT
молярная часть CO2
Compress_Coeff: FLOAT
коэффициент сжимаемости
Press_Drop: FLOAT
перепад давления [МПа]
Pipe_D: FLOAT
диаметр трубы [мм] при t=20C
Temp_Coeff_Pipe_Expan: FLOAT
коэффициент линейного теплового расширения материала трубы [1/C]
Pipe_Asp: FLOAT
шероховатость внутренней поверхности трубы [мм]
Diaphragm_D: FLOAT
диаметр диафрагмы [мм] при t=20C
Temp_Coeff_Diaphragm_Expan: FLOAT
коэффициент линейного теплового расширения материала диафрагмы [1/C]
Diaphragm_Calibr_Period: FLOAT
период калибровки диафрагмы [год]
Init_Circuit_R: FLOAT
начальный радиус входной кромки диафрагмы [мм]

Выходы:

Error_Code: INTEGER
код ошибки: 0 - нет ошибки, 1 - неверные входные аргументы
Gas_Density: FLOAT
плотность газа в рабочих условиях [кг/м³]
Dyn_Viscos: FLOAT
динамическая вязкость [mкПа * sec]
Adiabate_Degr: FLOAT
степень (показатель) адиабаты
Flowrate_Norm: FLOAT
объемный расход, приведенный к стандартным условиям (нормальный
объемный расход) [м³/час]
Mass_Flowrate: FLOAT
массовый расход, приведенный к стандартным условиям (нормальный
массовый расход) [кг/час]

Расчет расхода газа при прохождении через сужающее устройство с фланцевым отбором давления согласно ГОСТ 8.563.2-97

GasFlowrate_DF(IN1, IN2, ..., OUT1, OUT2, ...)

Входы:

Gas_Temp: FLOAT
температура газа [K]
Gas_Press: FLOAT
давление газа [МПа]
Gas_Density: FLOAT
плотность газа [кг/м³] при нормальных условиях (p=1atm, t=20C)
Molar_N2: FLOAT
молярная часть N2
Molar_CO2: FLOAT
молярная часть CO2
Compress_Coeff: FLOAT
коэффициент сжимаемости
Press_Drop: FLOAT
перепад давления [МПа]
Pipe_D: FLOAT
диаметр трубы [мм] при t=20C
Temp_Coeff_Pipe_Expan: FLOAT
коэффициент линейного теплового расширения материала трубы [1/C]
Pipe_Aspirity: FLOAT
шероховатость внутренней поверхности трубы [мм]
Diaphragm_D: FLOAT
диаметр диафрагмы [мм] при t=20C
Temp_Coeff_Diaphragm_Expan: FLOAT
коэффициент линейного теплового расширения материала диафрагмы [1/C]
Diaphragm_Calibr_Period: FLOAT
период калибровки диафрагмы [год]
Init_Circuit_R: FLOAT
начальный радиус входной кромки диафрагмы [мм]

Выходы:

Error_Code: INTEGER
код ошибки: 0 - нет ошибки, 1 - неверные входные аргументы
Gas_Density_Work: FLOAT
плотность газа в рабочих условиях [кг/м³]
Dyn_Viscosity: FLOAT
динамическая вязкость [mкПа * sec]
Adiabate_Degree: FLOAT
степень (показатель) адиабаты
Flowrate_Norm: FLOAT
объемный расход, приведенный к стандартным условиям (нормальный
объемный расход) [м³/час]
Mass_Flowrate_Norm: FLOAT
массовый расход, приведенный к стандартным условиям (нормальный
массовый расход) [кг/час]

Расчет расхода газа при прохождении через сужающее устройство с трехрадиусным отбором давления согласно ГОСТ 8.563.2-97

GasFlowrate_DT(IN1, IN2, ..., OUT1, OUT2, ...)

Входы:

Gas_Temp: FLOAT
температура газа [K]
Gas_Press: FLOAT
давление газа [МПа]
Gas_Density: FLOAT
плотность газа [кг/м³] при нормальных условиях (p=1atm, t=20C)
Molar_N2: FLOAT
молярная часть N2
Molar_CO2: FLOAT
молярная часть CO2
Compress_Coeff: FLOAT
коэффициент сжимаемости
Press_Drop: FLOAT
перепад давления [МПа]
Pipe_D: FLOAT
диаметр трубы [мм] при t=20C
Temp_Coeff_Pipe_Expan: FLOAT
коэффициент линейного теплового расширения материала трубы [1/C]
Pipe_Aspirity: FLOAT
шероховатость внутренней поверхности трубы [мм]
Diaphragm_D: FLOAT
диаметр диафрагмы [мм] при t=20C
Temp_Coeff_Diaphragm_Expan: FLOAT
коэффициент линейного теплового расширения материала диафрагмы [1/C]
Diaphragm_Calibr_Period: FLOAT
период калибровки диафрагмы [год]
Init_Circuit_R: FLOAT
начальный радиус входной кромки диафрагмы [мм]

Выходы:

Error_Code: INTEGER
код ошибки: 0 - нет ошибки, 1 - неверные входные аргументы
Gas_Density_Work: FLOAT
плотность газа в рабочих условиях [кг/м³]
Dyna_Viscosity: FLOAT
динамическая вязкость [mkPa * sec]
Adiabate_Degree: FLOAT
степень (показатель) адиабаты
Flowrate_Norm: FLOAT
объемный расход, приведенный к стандартным условиям (нормальный
объемный расход) [м³/час]
Mass_Flowrate_Norm: FLOAT
массовый расход, приведенный к стандартным условиям (нормальный
массовый расход) [кг/час]

Gas_Flow(IN1, IN2, ..., OUT1, OUT2, ...)

Входы:

Параметры течения газа:

T: FLOAT

температура газа [K]

P: FLOAT

давление газа [МПа]

Ros: FLOAT

плотность газа [кг/м³] при p=1atm, t=20C

N2: FLOAT

молярная часть N2

CO2: FLOAT

молярная часть CO2

dP: FLOAT

перепад давления [МПа]

K: FLOAT

коэффициент сжимаемости

Параметры измерительного трубопровода:

Dp20: FLOAT

диаметр ИТ [мм] при t=20C

Ap0, Ap1, Ap2: FLOAT

коэффициенты полинома для расчета температурного коэффициента
линейного расширения ИТ

sR: BOOLEAN

переключатель данных на ножке R:

TRUE - эквивалентная шероховатость Rsh

FALSE - отклонение профиля шероховатости Ra

R: FLOAT

шероховатость [мм] (задается переключателем sR)

Параметры сужающего устройства:

Type: INTEGER

тип СУ из следующего списка:

1 - диафрагма с угловым отбором давления

2 - диафрагма с фланцевым отбором давления

3 - диафрагма с трехрадиусным отбором давления

4 - сопло ИСА 1932

5 - эллипсное сопло

6 - сопло Вентури

7 - труба Вентури необработанная

8 - труба Вентури обработанная

9 - труба Вентури сварная из листовой стали

Dd20: FLOAT

диаметр СУ [мм] при t=20C

Ad0, Ad1, Ad2: FLOAT

Коэффициенты полинома для расчета температурного коэффициента
линейного расширения СУ

Rn: FLOAT

начальный радиус закругления входной кромки [мм]

sTau: BOOLEAN

переключатель данных на ножке Tau:

TRUE - период поверки СУ

FALSE - время от последней поверки СУ

Tau: FLOAT

время [год] (задается переключателем sTau)

Параметры расчета:

Nmax: INTEGER

максимальное количество итераций расчета, 0 - не ограничено

Выходы:

Ret: INTEGER

код возврата:

0 - вычисления выполнены без замечаний

1 - условия ГОСТ нарушены, но расчет выполнен

2 - условия ГОСТ нарушены, расчет НЕ выполнен

Qc: FLOAT

объемный нормальный расход [нм³/сек]

Qv: FLOAT

объемный рабочий расход [м³/сек]

Qm: FLOAT

массовый расход [кг/сек]

Расчет расхода газа через сужающее устройство (СУ) ГОСТ 8.586.1,2,3,4,5-2005 с расширенным набором выходов

Gas_FlowExt(IN1, IN2, ..., OUT1, OUT2, ...)

Входы:

Параметры течения газа:

T: FLOAT

температура газа [K]

P: FLOAT

давление газа [МПа]

Ros: FLOAT

плотность газа [кг/м³] при p=1atm, t=20C

N2: FLOAT

молярная часть N2

CO2: FLOAT

молярная часть CO2

dP: FLOAT

перепад давления [МПа]

K: FLOAT

коэффициент сжимаемости

Параметры измерительного трубопровода:

Dp20: FLOAT

диаметр ИТ [мм] при t=20C

Ap0, Ap1, Ap2: FLOAT

коэффициенты полинома для расчета температурного коэффициента линейного расширения ИТ

sR: BOOLEAN

переключатель данных на ножке R:

TRUE - эквивалентная шероховатость Rsh

FALSE - отклонение профиля шероховатости Ra

R: FLOAT

шероховатость [мм] (задается переключателем sR)

Параметры сужающего устройства:

Type: INTEGER

тип СУ из следующего списка:

1 - диафрагма с угловым отбором давления

2 - диафрагма с фланцевым отбором давления

3 - диафрагма с трехрадиусным отбором давления

4 - сопло ИСА 1932

5 - эллипсное сопло

6 - сопло Вентури

7 - труба Вентури необработанная

8 - труба Вентури обработанная

9 - труба Вентури сварная из листовой стали

Dd20: FLOAT

диаметр СУ [мм] при $t=20^{\circ}\text{C}$

Ad0, Ad1, Ad2: FLOAT

Коэффициенты полинома для расчета температурного коэффициента линейного расширения СУ

Rn: FLOAT

начальный радиус закругления входной кромки [мм]

sTau: BOOLEAN

переключатель данных на ножке Tau:

TRUE - период поверки СУ

FALSE - время от последней поверки СУ

Tau: FLOAT

время [год] (задается переключателем sTau)

Параметры расчета:

Nmax: INTEGER

максимальное количество итераций расчета, 0 - не ограничено

Выходы:

Ret: INTEGER

код возврата:

0 - вычисления выполнены без замечаний

1 - условия ГОСТ нарушены, но расчет выполнен

2 - условия ГОСТ нарушены, расчет НЕ выполнен

Qc: FLOAT

объемный нормальный расход [нм³/сек]

Qv: FLOAT

объемный рабочий расход [м³/сек]

Qm: FLOAT

массовый расход [кг/сек]

Ro: FLOAT

плотность газа в рабочих условиях [кг/м³]

Mu: FLOAT

динамическая вязкость [мкПа*сек]
 Kp: FLOAT
 показатель степени адиабаты
 Ap: FLOAT
 температурный коэфф. линейного расширения ИТ [1e-6/C]
 Dp: FLOAT
 диаметр ИТ в рабочих условиях [мм]
 Ad: FLOAT
 температурный коэфф. линейного расширения СУ [1e-6/C]
 Dd: FLOAT
 диаметр СУ в рабочих условиях [мм]
 betta: FLOAT
 относительный диаметр
 E: FLOAT
 коэффициент скорости входа
 C: FLOAT
 коэффициент истечения
 DKp: FLOAT
 коэффициент, учитывающий затупление кромки диафрагмы
 eps: FLOAT
 коэффициент расширения
 Ksh: FLOAT
 коэффициент шероховатости
 Re: FLOAT
 число Рейнольдса
 N: INTEGER
 количество выполненных итераций

3.11 Работа с дискретным сигналом

Работа с дискретным сигналом:

TS_01(I, X, S, CD, R, J, M, E)

Входы:

I: INTEGER
 feedback-параметр
 X: BOOLEAN
 срабатывание датчика
 S: INTEGER
 статус датчика
 CD: BOOLEAN
 отсутствие связи
 R: BOOLEAN
 режим работы: включить анализ наличия связи

Выходы:

J: INTEGER
 feedback-параметр
 M: BOOLEAN
 необходимость формирования сообщения
 E: BOOLEAN

необходимость формирования исполняющего воздействия

3.12 Вызов блоков и схем, разработанных пользователем

Вызов блоков и схем, разработанных пользователем, осуществляется по имени, параметры, если они есть, заключаются в скобки, перечисляются через запятую. В Списке параметров сначала перечисляются входные параметры, затем – выходные. В качестве входных параметров допустимо использовать переменные, константы или вызовы функций. Для выходных параметров допустимо использовать только переменные.

[illegible][illegible]