

ШПАРГАЛКА ПО CODESYS V3.5

КОММЕНТАРИИ

// однострочный комментарий

(* многострочный
 комментарий *)

ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ

<Имя> : <Тип> := < начальное_значение>;

Требования к именам переменных

Если начальное значение не указано, то используется «нулевое». Пробелы до и после «:», «:=» и «;» не являются значимыми.

Примеры:

```
iVar: INT := 123;
rVar: REAL := 11.22;
wVar: WORD := 16#FFAB;
// максимальная длина строки – 20 символов.
sVar: STRING (20) := 'hello, world';
```

ОСНОВНЫЕ ОБЛАСТИ ПЕРЕМЕННЫХ

Область	Назначение	Где создаются
VAR ... END VAR	Локальные переменные	В программах, функциональных блоках, функциях, методах
VAR INPUT ... END VAR	Входные переменные	
VAR OUTPUT ... END VAR	Выходные переменные	
VAR IN OUT ... END VAR	Входы-выходы	
VAR CONSTANT ... END VAR	Константы	
VAR GLOBAL ... END VAR	Глобальные переменные	В объекте Список глобальных переменных
VAR RETAIN ... END VAR	RETAIN-переменные (энергонезависимые)	В программах и списках глобальных переменных
VAR_GLOBAL RETAIN ... END_VAR	Глобальные RETAIN-переменные (энергонезависимые)	В объекте Список глобальных переменных
VAR_GLOBAL PERSISTENT RETAIN ... END_VAR	Глобальные PERSISTENT-переменные (энергонезависимые)	В объекте Persistent-переменные

ТИПЫ ДАННЫХ

Логические и целочисленные типы данных

Тип	Размер (в байтах)	Диапазон возможных значений
BOOL	1	TRUE и FALSE
BYTE/USINT	1	0...255
WORD/UINT	2	0...65535
DWORD/UDINT	4	0...2 ³² -1
LWORD/ULINT	8	0...2 ⁶⁴ -1
SINT	1	-128...127
INT	2	-32768...32767
DINT	4	-2 ³¹ ...2 ³¹ -1
LINT	8	-2 ⁶³ ...2 ⁶³ -1

Примечания:

1. Поддерживается [побитовый доступ](#):

```
// Запись TRUE в 15-й бит переменной типа WORD
wVar.15 := TRUE;
```

Значения с плавающей точкой

Тип	Размер (в байтах)	Точность (число значащих цифр после точки)
REAL	4	7-8
LREAL	8	15-17
Примеры	11.22 1.25E3 (в экспоненциальной нотации)	

Строки

Тип	Размер символа (в байтах)	Кодировка	Пример литерала (внимание на кавычки)
STRING	1	ASCII-based (например, Win1251)	sVar: STRING (20) := 'hello, world';
WSTRING	2	UCS2 (вариант UTF-16)	wsVar: WSTRING (20) := "привет, мир";

Примечания:

1. Строки в CODESYS являются [нуль-терминированными](#), и при их объявлении автоматически резервируется память для символа-терминатора. Т.е. STRING(20) займет 21 байт памяти, а WSTRING(20) – 42.
2. При объявлении можно указать максимальную длину строки. Явных ограничений для этого нет, но см. врезку **БИБЛИОТЕКА STANDARD** и примечания к ней.
3. Для отображения кириллицы в визуализации следует использовать только тип WSTRING.

Дата и время

Тип	Размер (в байтах)	Диапазон	Описание	Дискретность
TIME	4	T#0ms...T#49d17h2m47s295ms	Интервал времени	Миллисекунды
DT	4	DT#1970-1-1-0:0:0... DT#2106-02-07-06:28:15	Дата и время	Секунды
DATE	4	DATE#1970-1-1-... DT#2106-02-07	Дата	Секунды (хотя время суток и не обрабатывается)
TOD	4	TOD#00:00:00... TOD#23:59:59.999	Время суток	Миллисекунды

Примечания:

1. Возможные арифметические операции над этими типами:

TIME + TIME = TIME

TIME – TIME = TIME

TIME · целочисленный тип = TIME

DT + TIME = DT

DT – TIME = DT

DT – DT = TIME

TOD + TIME = TOD

TOD – TIME = TOD

2. Типы LTIME/LDT/LDATE/LTOD отличаются размером (8 байт), диапазоном и дискретностью (наносекунды).

ПРЕОБРАЗОВАНИЯ ТИПОВ

```
TO_<тип>
// преобразуем число с плавающей точкой в строку
sVar := TO_STRING(rVar);
// преобразуем целочисленную переменную в интервал времени (число миллисекунд)
tVar := TO_TIME(udiVar);
// с другими типами – по аналогии
```

СОСТАВНЫЕ ТИПЫ ДАННЫХ (см. детали в справке CODESYS и [этой статье](#))

[массивы](#)

[структуры](#)

[объединения](#)

[перечисления](#)

НЕМНОГО ПРО МАССИВЫ (как самый активно используемый составной тип данных)

```
// Объявление одномерного массива из четырех элементов с начальными значениями 10, 20, 30 и
// 40 соответственно
VAR
  aiData: ARRAY [0..3] OF INT := [10, 20, 30, 40];
END_VAR
// Обращение к элементу массива в коде программы
// iVar получит значение 10
iVar := aiData[0];
// записываем в элемент массива с индексом 1 значение 25
aiData[1] := 25;
```

ОСНОВНЫЕ ОПЕРАТОРЫ

Мнемоника в ST	Мнемоника в графических языках	Описание
Арифметические		
:=	MOVE	Присваивание значений одной переменной (или литерала) другой. В ST присваивание выполняется «справа налево», в графических языках – слева направо
+	ADD	Сложение
-	SUB	Вычитание
*	MUL	Умножение
/	DIV	Целочисленное деление
MOD		Получение остатка от целочисленного деления
Побитовые (применимы для BOOL и всех целочисленных типов)		
NOT		Отрицание
AND		Побитовое «И»
OR		Побитовое «ИЛИ»
XOR		Побитовое «Исключающее ИЛИ»
Операторы сравнения		
=	EQ	Проверка на равенство
<>	NE	Проверка на неравенство
>	GT	Проверка на «первое значение больше второго»
<	LT	Проверка на «первое значение меньше второго»
>=	GE	Проверка на «первое значение больше или равно второму»
<=	LE	Проверка на «первое значение меньше или равно второму»

Битовые сдвиги	
SHL	Сдвиг влево с дополнением нулями
SHR	Сдвиг вправо с дополнением нулями
ROL	Циклический сдвиг влево («по кругу»)
ROR	Циклический сдвиг вправо («по кругу»)
Операторы выбора	
SEL	Выбор из двух значений совпадающих типов
MUX	Выбор из произвольного числа значений совпадающих типов
LIMIT	Ограничение диапазона значений
Математические операторы	
MIN	Выбор минимума из произвольного числа значений совпадающих типов
MAX	Выбор максимума из произвольного числа значений совпадающих типов
ABS	Значение по модулю
SQRT	Извлечение квадратного корня
LN	Натуральный логарифм
LOG	Десятичный логарифм
EXP	Показательная функция (экспонента)
EXPT	Возведение в степень
TRUNC	«Усечение» значения типа REAL до DINT с отбрасыванием дробной части
TRUNC_INT	«Усечение» значения типа REAL до INT с отбрасыванием дробной части
Тригонометрические операторы	
SIN	Синус
COS	Косинус
TAN	Тангенс
ASIN	Арксинус
ACOS	Арккосинус
ATAN	Арктангенс
TIME	Значение системного тика (время с момента запуска контроллера в миллисекундах)
SIZEOF, XSIZEOF	Определение размера объекта в байтах
<i>Примечания:</i>	
1. Порядок выполнения операторов в выражении определяется их приоритетом и расстановкой скобок	

УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ ЯЗЫКА ST

Оператор	Описание
IF...THEN...ELSIF...THEN...ELSE...END IF	Оператор условного выбора (проверка набора условий и выполнение операций для того из них, которое первым оказалось истинным)
CASE...OF...:...ELSE...END CASE	Оператор многозначного выбора (проверка целочисленной переменной на совпадение с набором значений и выполнение операций в случае совпадения)
FOR...TO...BY...DO...END FOR	Цикл с заданным числом итераций
WHILE...DO...END WHILE	Цикл с предусловием (проверка условия выполняется до итерации)
REPEAT...UNTIL...END REPEAT	Цикл с постусловием (проверка условия выполняется после итерации)
CONTINUE	Переход к следующей итерации цикла
EXIT	Выход из цикла
RETURN	Выход из POU

ОСНОВНЫЕ ТИПЫ ПРОГРАММНЫХ ОБЪЕКТОВ (POU)

Название	Наличие собственной памяти и сохранение значений переменных между вызовами	Объявление экземпляров
Функция (FUN)	Нет	Нет
Функциональный блок (ФБ)	Да	Да
Программа (PRG)	Да	Нет

БИБЛИОТЕКА STANDARD (базовые функции и функциональные блоки)

Название	Описание
Строковые функции	
CONCAT	Объединение строк
DELETE	Удаление подстроки из строки
FIND	Поиск подстроки в строке
INSERT	Вставка подстроки в строку
LEFT	Выделение левой части строки заданной длины
LEN	Определение длины строки
MID	Выделение произвольной части строки заданной длины
REPLACE	Замена подстроки в строке
RIGHT	Выделение правой части строки заданной длины

Примечания:

1. Функции для работы со строками типа WSTRING и таймеры с типом LTIME доступны в библиотеке [Standard64](#)
2. Другие часто используемые блоки (линейное масштабирование, генератор импульсов, BLINK, ПИД-регулятор и т.д.) доступны в библиотеке [Util](#)
3. Строковые функции поддерживают только обработку строк, длина которых не превышает 255 символов. Для работы с более длинными строками используйте функции библиотеки [StringUtils](#)
4. Расширенные функции для работы со строками (парсинг, форматирование и т.д.) доступны в библиотеке [OwenStringUtils](#)

Детекторы фронта

R_TRIG	Генерирует единичный импульс по переднему фронту (R_TRIG) или заднему фронту (F_TRIG) заданной булевой переменной
F_TRIG	

Триггеры

SR	Переключатель с доминантой включения
RS	Переключатель с доминантой отключения

Счетчики

CTU	Инкрементный счетчик импульсов
CTD	Декрементный счетчик импульсов
CTUD	Счетчик импульсов с поддержкой обоих направлений счёта

Таймеры

TP	Генератор логического сигнала заданной длительности
TON	Таймер с задержкой на включение
TOF	Таймер с задержкой на отключение

ДРУГИЕ ПОЛЕЗНЫЕ БИБЛИОТЕКИ ИЗ ДИСТРИБУТИВА CODESYS

Библиотека	Назначение	Документация на русском
CAA Memory	Работа с памятью (копирование блоков памяти, их сравнение, заполнение и т.д.)	ОТКРЫТЬ
CAA File	Работы с файлами (чтение/запись)	ОТКРЫТЬ
CAA SerialCom	Работа с COM-портом для реализации нестандартных протоколов обмена	ОТКРЫТЬ
CAA NetBaseServices	Работа с TCP/UDP для реализации нестандартных протоколов обмена	ОТКРЫТЬ
Visu Utils	Работы с визуализацией из кода программы (переключение экранов, открытие окон и т.д.)	ОТКРЫТЬ

ПОЛЕЗНЫЕ ССЫЛКИ

[Онлайн-справка CODESYS](#)

[Документация](#), [библиотеки](#), [примеры](#) и [видео](#) по CODESYS от компании OBEH

[Онлайн-курс по CODESYS V3.5 на платформе Stepik](#)

Статьи:
[«Отладка проектов в CODESYS V3»](#)
[«Работа с памятью в CODESYS V3»](#)
[«Использование задач в CODESYS V3»](#)
- [другие статьи с портала oscat.ru](#)

[Руководство по кодированию](#) от консорциума [PLCopen](#)