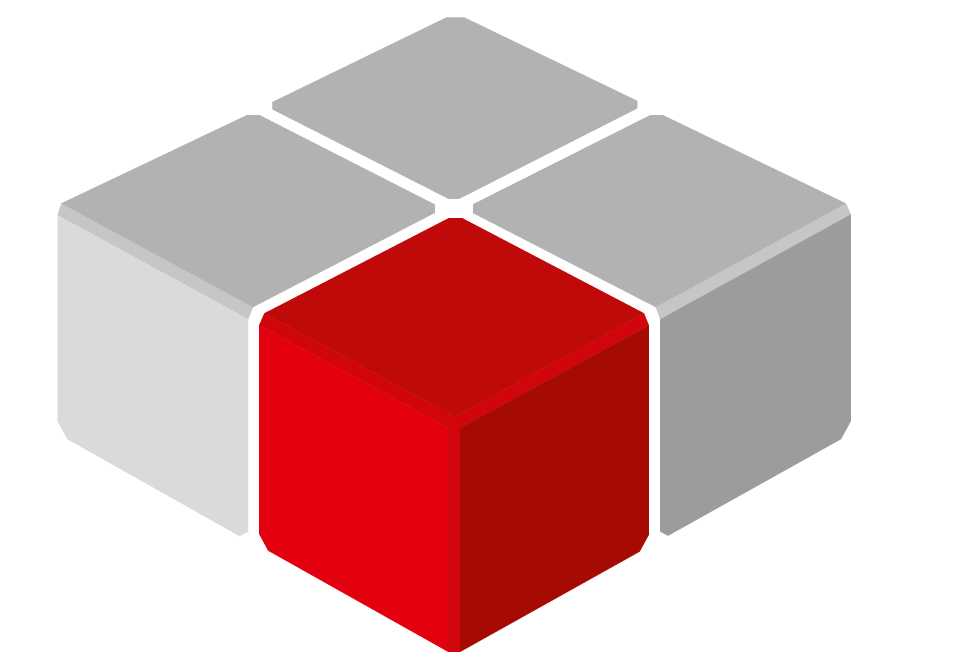


**CODESYS V3.5**

**Описание библиотеки** **OwenStringUtils**



Руководство пользователя

20.06.2020

версия 2.1

Оглавление

[1 Цель документа 4](#_Toc44667378)

[2 Описание библиотеки OwenStringUtils 5](#_Toc44667379)

[2.1 Установка библиотеки 5](#_Toc44667380)

[2.2 Добавление библиотеки в проект CODESYS 6](#_Toc44667381)

[2.3 Описание библиотеки 7](#_Toc44667382)

[2.3.1 Функция CP1251\_TO\_UNICODE 7](#_Toc44667383)

[2.3.2 Функция UNICODE\_TO\_CP1251 8](#_Toc44667384)

[2.3.3 Функция Before 9](#_Toc44667385)

[2.3.4 Функция WBefore 10](#_Toc44667386)

[2.3.5 Функция After 11](#_Toc44667387)

[2.3.6 Функция WAfter 12](#_Toc44667388)

[2.3.7 Функция Between 13](#_Toc44667389)

[2.3.8 Функция WBetween 14](#_Toc44667390)

[2.3.9 Функция LowerCase 15](#_Toc44667391)

[2.3.10 Функция WLowerCase 16](#_Toc44667392)

[2.3.11 Функция UpperCase 17](#_Toc44667393)

[2.3.12 Функция WUpperCase 18](#_Toc44667394)

[2.3.13 Функция REAL\_TO\_STRING\_FORMAT 19](#_Toc44667395)

[2.3.14 Функция LREAL\_TO\_STRING\_FORMAT 20](#_Toc44667396)

[2.3.15 Функция DT\_TO\_STRING\_FORMAT 21](#_Toc44667397)

[2.3.16 Функция DATE\_TO\_STRING\_FORMAT 22](#_Toc44667398)

[2.3.17 Функция TOD\_TO\_STRING\_FORMAT 23](#_Toc44667399)

[2.3.18 Функция FindSubstringPosAfterN 24](#_Toc44667400)

[2.3.19 Функция WFindSubstringPosAfterN 25](#_Toc44667401)

[2.3.20 Функция ReplaceSubstring 26](#_Toc44667402)

[2.3.21 Функция WReplaceSubstring 27](#_Toc44667403)

[2.3.22 Функция ReplaceAllSubstrings 28](#_Toc44667404)

[2.3.23 Функция WReplaceAllSubstrings 29](#_Toc44667405)

[2.3.24 Функция CONCAT4 30](#_Toc44667406)

[2.3.25 Функция WCONCAT4 31](#_Toc44667407)

[2.3.26 Функция CONCAT8 32](#_Toc44667408)

[2.3.27 Функция WCONCAT8 33](#_Toc44667409)

[2.3.28 Функция ADD\_CHAR 34](#_Toc44667410)

[2.3.29 Функция WADD\_CHAR 35](#_Toc44667411)

[2.3.30 Функция HEX\_STR\_TO\_WORD 36](#_Toc44667412)

[2.3.31 Функция WORD\_TO\_HEX\_STR 37](#_Toc44667413)

[2.3.32 Функция BYTES\_TO\_IPSTRING 38](#_Toc44667414)

[2.3.33 Функция IPSTRING\_TO\_BYTES 39](#_Toc44667415)

[2.3.34 Функция UDINT\_TO\_IPSTRING 40](#_Toc44667416)

[2.3.35 Функция IPSTRING\_TO\_UDINT 41](#_Toc44667417)

[2.3.36 Функция MAC\_TO\_STRING 42](#_Toc44667418)

[3 Приложение А. Заполнители формата времени 43](#_Toc44667419)

# Цель документа

Настоящее руководство представляет собой описание библиотеки **OwenStringUtils**, которая предоставляет пользователю дополнительный функционал для работы со строками – в частности, функции конвертации строк **ASCII** в строки **Unicode** и **Unicode** в **ASCII**. В данном документе описана версия библиотеки **3.5.4.8**.

|  |  |
| --- | --- |
|  | **ПРИМЕЧАНИЕ**  Функции библиотеки позволяют работать со строками, длина которых не превышает **255** символов. |

|  |  |
| --- | --- |
|  | **ПРИМЕЧАНИЕ**  Базовые функции работы со строками содержатся в библиотеках **Standard**, **Standard64** и **StringUtils**, которые входят в состав **CODESYS**. |

# Описание библиотеки OwenStringUtils

## Установка библиотеки

Для установки библиотеки в **CODESYS** в меню **Инструменты** следует выбрать пункт **Репозиторий библиотек**, нажать кнопку **Установить,** указать путь к библиотеке и нажать **Открыть**:

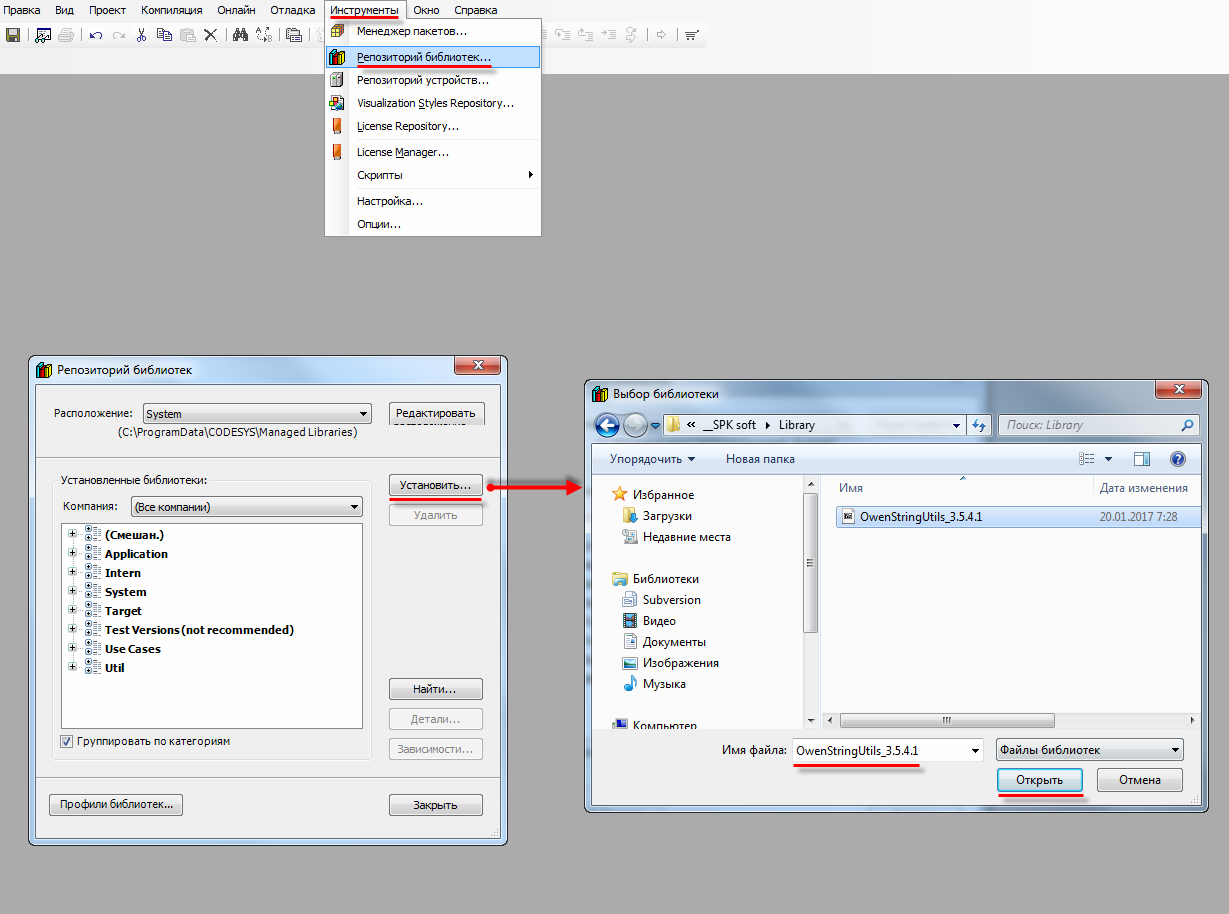


Рисунок 2.1 – Установка библиотеки в среду CODESYS

## Добавление библиотеки в проект CODESYS

Для добавления библиотеки **OwenStringUtils** в проект **CODESYS** следует в **Менеджере библиотек** нажать кнопку **Добавить библиотеку** и в строке поиска ввести **OwenStringUtils**, после чего выбрать из списка нужную библиотеку и нажать **ОК**.

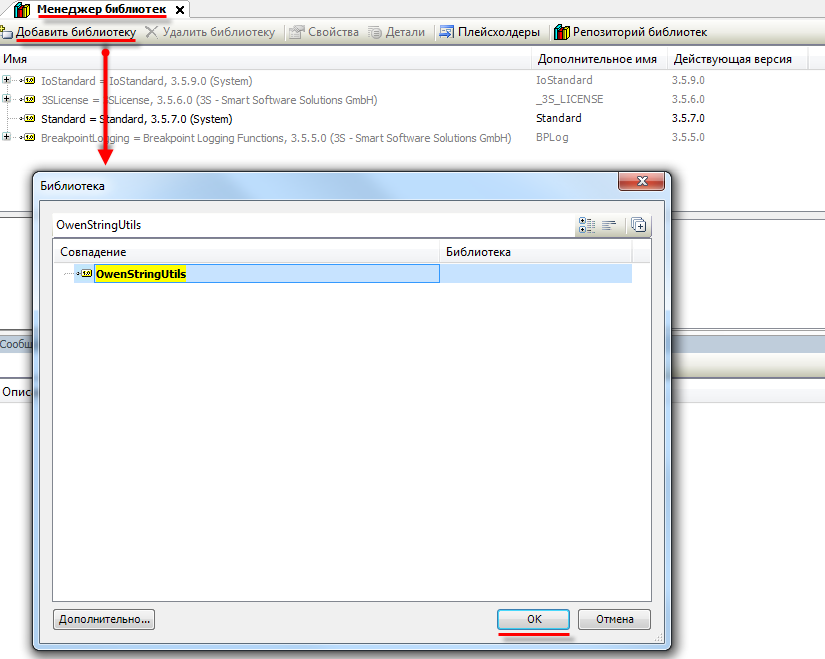
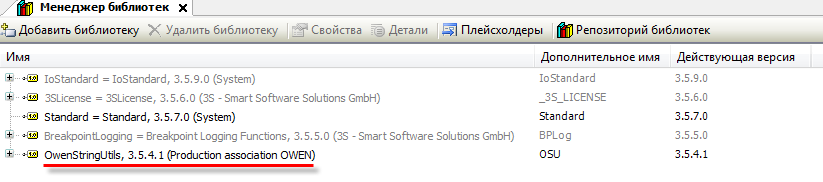


Рисунок 2.2 – Добавление библиотеки OwenStringUtils

После добавления библиотека появится в списке **Менеджера библиотек**:

Рисунок 2.3 – Список библиотек проекта

|  |  |
| --- | --- |
|  | **ПРИМЕЧАНИЕ**  При обращении к функциям библиотеки следует перед их названием указывать префикс **OSU** (пример: **OSU.After**). |

## Описание библиотеки

### Функция CP1251\_TO\_UNICODE

Функция **CP1251\_TO\_UNICODE** используется для конвертации переменной типа **STRING**, содержащей строку в кодировке [ASCII](https://ru.wikipedia.org/wiki/ASCII) ([CP1251](https://ru.wikipedia.org/wiki/Windows-1251)), в переменную типа **WSTRING**, содержащую строку в кодировке [Unicode](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4) ([UCS-2](http://www.unicode.org/faq/utf_bom.html#utf16-11)).

**Таблица 2.1 – Описание входов и выходов функции CP1251\_TO\_UNICODE**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sInputString | STRING(255) | Исходная строка в кодировке [ASCII](https://ru.wikipedia.org/wiki/ASCII) ([CP1251](https://ru.wikipedia.org/wiki/Windows-1251)) |
| **Выходные переменные** | | |
| CP1251\_TO\_UNICODE | WSTRING(255) | Строка в кодировке [Unicode](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4) ([UCS-2](http://www.unicode.org/faq/utf_bom.html#utf16-11)) |

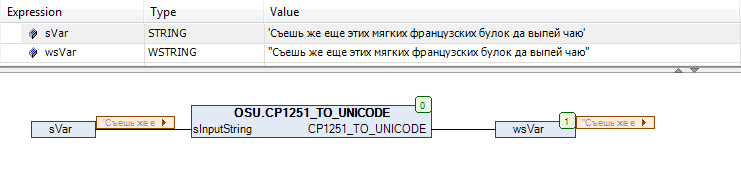


Рисунок 2.4 – Пример использования функции CP1251\_TO\_UNICODE на языке CFC

### Функция UNICODE\_TO\_CP1251

Функция **UNICODE\_TO\_CP1251** используется для конвертации переменной типа **WSTRING**, содержащей строку в кодировке [Unicode](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4) ([UCS-2](http://www.unicode.org/faq/utf_bom.html#utf16-11)), в переменную типа **STRING**, содержащую строку в кодировке [ASCII](https://ru.wikipedia.org/wiki/ASCII) ([CP1251](https://ru.wikipedia.org/wiki/Windows-1251))

**Таблица 2.2 – Описание входов и выходов функции UNICODE\_TO\_CP1251**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsInputString | WSTRING(255) | Исходная строка в кодировке [Unicode](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4) ([UCS-2](http://www.unicode.org/faq/utf_bom.html#utf16-11)) |
| **Выходные переменные** | | |
| UNICODE\_TO\_CP1251 | STRING(255) | Строка в кодировке [ASCII](https://ru.wikipedia.org/wiki/ASCII) ([CP1251](https://ru.wikipedia.org/wiki/Windows-1251)) |

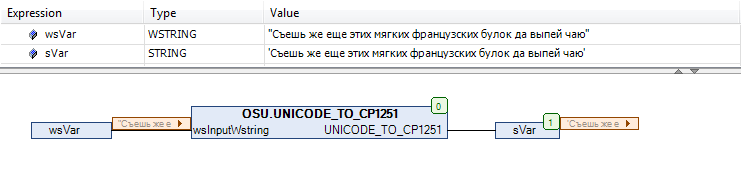


Рисунок 2.5 – Пример использования функции UNICODE\_TO\_CP1251 на языке CFC

### Функция Before

Функция **Before** возвращает фрагмент исходной строки **sSource**, предшествующий первому вхождению подстроки **sPostfix** (не включая саму подстроку). Все переменные функции имеют тип **STRING**.

**Таблица 2.3 – Описание входов и выходов функции Before**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sSource | STRING(255) | Исходная строка |
| sPostfix | STRING(255) | Подстрока |
| **Выходные переменные** | | |
| Before | STRING(255) | Фрагмент исходной строки, предшествующий первому вхождению подстроки (не включая саму подстроку) |

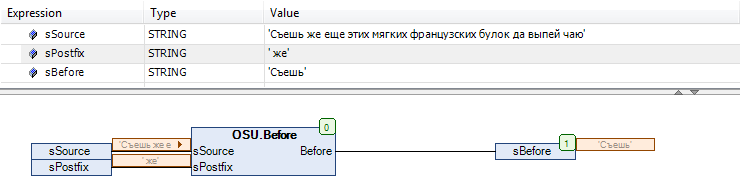


Рисунок 2.6 – Пример использования функции Before на языке CFC

### Функция WBefore

Функция **WBefore** возвращает фрагмент исходной строки **wsSource**, предшествующий первому вхождению подстроки **wsPostfix** (не включая саму подстроку). Все переменные функции имеют тип **WSTRING**.

**Таблица 2.4 – Описание входов и выходов функции WBefore**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsSource | WSTRING(255) | Исходная строка |
| wsPostfix | WSTRING(255) | Подстрока |
| **Выходные переменные** | | |
| WBefore | WSTRING(255) | Фрагмент исходной строки, предшествующий первому вхождению подстроки (не включая саму подстроку) |

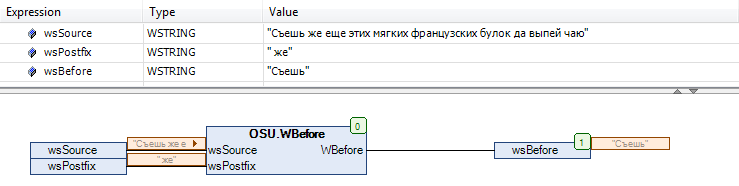


Рисунок 2.7 – Пример использования функции WBefore на языке CFC

### Функция After

Функция **After** возвращает фрагмент исходной строки **sSource**, следующий за первым вхождением подстроки **sPrefix** (не включая саму подстроку). Все переменные функции имеют тип **STRING**.

**Таблица 2.5 – Описание входов и выходов функции After**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sSource | STRING(255) | Исходная строка |
| sPrefix | STRING(255) | Подстрока |
| **Выходные переменные** | | |
| After | STRING(255) | Фрагмент исходной строки, следующий за первым вхождением подстроки (не включая саму подстроку) |

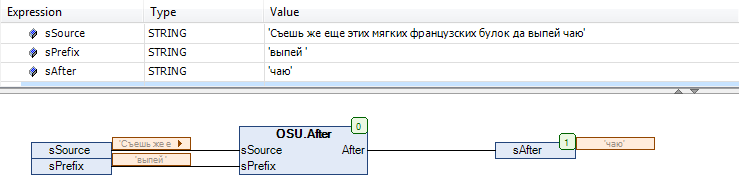


Рисунок 2.8 – Пример использования функции After на языке CFC

### Функция WAfter

Функция **WAfter** возвращает фрагмент исходной строки **wsSource**, следующий за первым вхождением подстроки **wsPrefix** (не включая саму подстроку). Все переменные функции имеют тип **WSTRING**.

**Таблица 2.6 – Описание входов и выходов функции WAfter**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsSource | WSTRING(255) | Исходная строка |
| wsPrefix | WSTRING(255) | Подстрока |
| **Выходные переменные** | | |
| WAfter | WSTRING(255) | Фрагмент исходной строки, следующий за первым вхождением подстроки (не включая саму подстроку) |

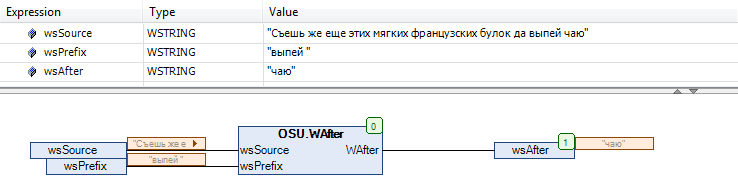


Рисунок 2.9 – Пример использования функции WAfter на языке CFC

### Функция Between

Функция **Between** возвращает фрагмент исходной строки **sSource**, расположенный между первыми вхождениями начальной подстроки **sPrefix** и конечной подстроки **sPostfix** (не включая сами подстроки). Все переменные функции имеют тип **STRING**.

**Таблица 2.7 – Описание входов и выходов функции Between**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sSource | STRING(255) | Исходная строка |
| sPrefix | STRING(255) | Начальная подстрока |
| sPostfix | STRING(255) | Конечная подстрока |
| **Выходные переменные** | | |
| Between | STRING(255) | Фрагмент исходной строки, расположенный между первыми вхождениями начальной и конечной подстрок (не включая сами подстроки) |

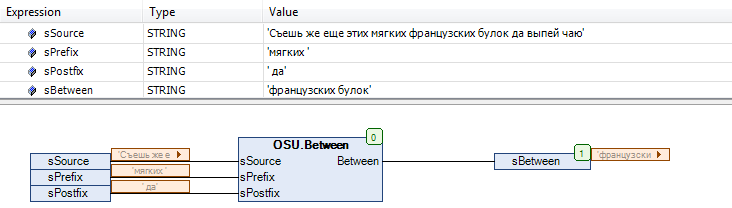


Рисунок 2.10 – Пример использования функции Between на языке CFC

### Функция WBetween

Функция **WBetween** возвращает фрагмент исходной строки **wsSource**, расположенный между первыми вхождениями начальной подстроки **wsPrefix** и конечной подстроки **wsPostfix** (не включая сами подстроки). Все переменные функции имеют тип **WSTRING**.

**Таблица 2.8 – Описание входов и выходов функции WBetween**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsSource | WSTRING(255) | Исходная строка |
| wsPrefix | WSTRING(255) | Начальная подстрока |
| wsPostfix | WSTRING(255) | Конечная подстрока |
| **Выходные переменные** | | |
| WBetween | WSTRING(255) | Фрагмент исходной строки, расположенный между первыми вхождениями начальной и конечной подстрок (не включая сами подстроки) |

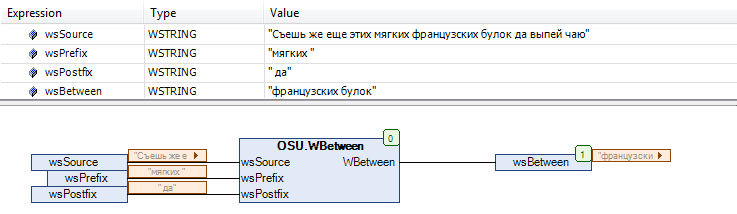


Рисунок 2.11 – Пример использования функции WBetween на языке CFC

### Функция LowerCase

Функция **LowerCase** преобразует все символы исходной строки **sStringToConvert** (в кодировке [СP1251](https://ru.wikipedia.org/wiki/Windows-1251)) в нижний регистр. Все переменные функции имеют тип **STRING**.

**Таблица 2.9 – Описание входов и выходов функции LowerCase**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sStringToConvert | STRING(255) | Исходная строка |
| **Выходные переменные** | | |
| LowerCase | STRING(255) | Строка в нижнем регистре |

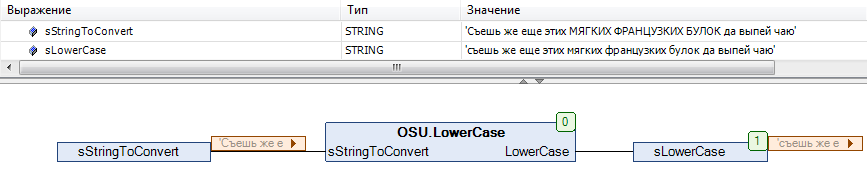


Рисунок 2.12 – Пример использования функции LowerCase на языке CFC

### Функция WLowerCase

Функция **WLowerCase** преобразует символы русского и английского алфавита исходной строки **wsStringToConvert** в нижний регистр. Все переменные функции имеют тип **WSTRING**.

**Таблица 2.10 – Описание входов и выходов функции WLowerCase**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsStringToConvert | WSTRING(255) | Исходная строка |
| **Выходные переменные** | | |
| WLowerCase | WSTRING(255) | Строка в нижнем регистре |

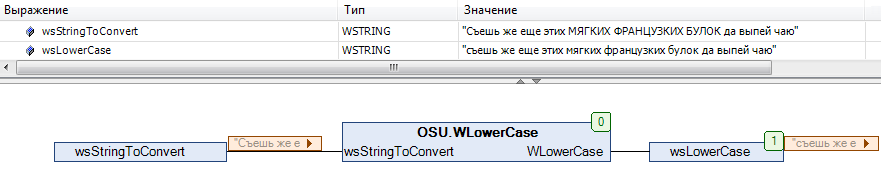


Рисунок 2.13 – Пример использования функции WLowerCase на языке CFC

### Функция UpperCase

Функция **UpperCase** преобразует все символы исходной строки **sStringToConvert** (в кодировке [СP1251](https://ru.wikipedia.org/wiki/Windows-1251))в верхний регистр. Все переменные функции имеют тип **STRING**.

**Таблица 2.11 – Описание входов и выходов функции UpperCase**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sStringToConvert | STRING(255) | Исходная строка |
| **Выходные переменные** | | |
| UpperCase | STRING(255) | Строка в верхнем регистре |

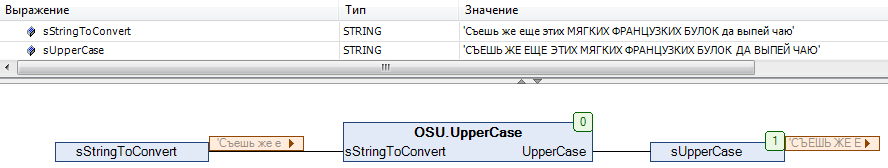


Рисунок 2.14 – Пример использования функции UpperCase на языке CFC

### Функция WUpperCase

Функция **WUpperCase** преобразует символы русского и английского алфавита исходной строки **wsStringToConvert** в верхний регистр. Все переменные функции имеют тип **WSTRING**.

**Таблица 2.12 – Описание входов и выходов функции WUpperCase**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsStringToConvert | WSTRING(255) | Исходная строка |
| **Выходные переменные** | | |
| WUpperCase | WSTRING(255) | Строка в верхнем регистре |

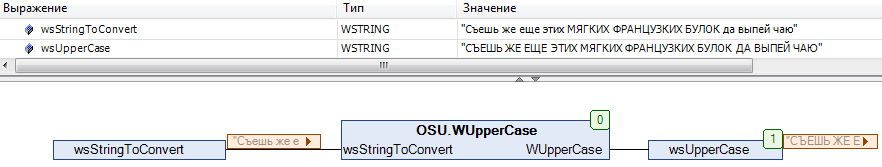


Рисунок 2.15 – Пример использования функции WUpperCase на языке CFC

### Функция REAL\_TO\_STRING\_FORMAT

Функция **REAL\_TO\_STRING\_FORMAT** преобразует значение с плавающей точкой типа **REAL** в форматированную строку типа **STRING** с настраиваемым символом-разделителем целой/ дробной части и количеством знаков после разделителя. Допустимые символы-разделители определяются перечислением **DECIMAL\_SEPARATOR**. В случае выбора недопустимого символа в качестве разделителя используется точка.

**Таблица 2.13 – Описание входов и выходов функции REAL\_TO\_STRING\_FORMAT**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| rValueToConvert | REAL | Значение с плавающей точкой |
| usiSignificantDigitsCount | USINT | Количество знаков после разделителя |
| eDecimalSeparator | DECIMAL\_SEPARATOR | Разделитель целой и дробной части |
| **Выходные переменные** | | |
| REAL\_TO\_STRING\_FORMAT | STRING(80) | Значение в виде форматированной строки |

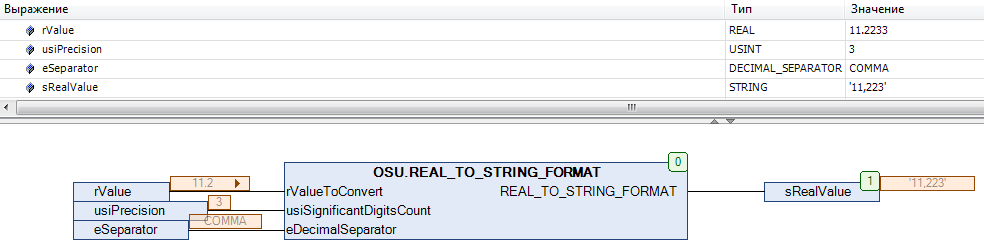


Рисунок 2.16 – Пример использования функции REAL\_TO\_STRING\_FORMAT на языке CFC

### Функция LREAL\_TO\_STRING\_FORMAT

Функция **LREAL\_TO\_STRING\_FORMAT** преобразует значение с плавающей точкой типа **LREAL** в форматированную строку типа **STRING** с настраиваемым символом-разделителем целой/ дробной части и количеством знаков после разделителя. Допустимые символы-разделители определяются перечислением **DECIMAL\_SEPARATOR**. В случае выбора недопустимого символа в качестве разделителя используется точка.

**Таблица 2.14 – Описание входов и выходов функции LREAL\_TO\_STRING\_FORMAT**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| lrValueToConvert | LREAL | Значение с плавающей точкой |
| usiSignificantDigitsCount | USINT | Количество знаков после разделителя |
| eDecimalSeparator | DECIMAL\_SEPARATOR | Разделитель целой и дробной части |
| **Выходные переменные** | | |
| LREAL\_TO\_STRING\_FORMAT | STRING(80) | Значение в виде форматированной строки |

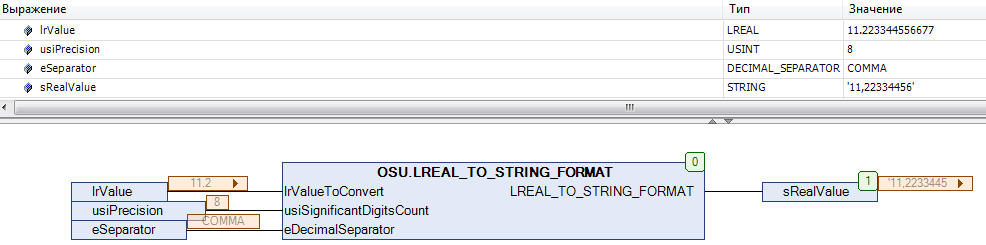


Рисунок 2.17 – Пример использования функции LREAL\_TO\_STRING\_FORMAT на языке CFC

### Функция DT\_TO\_STRING\_FORMAT

Функция **DT\_TO\_STRING\_FORMAT** заменяет в строке **sFormatString** первое вхождение подстроки типа %t[<заполнители>] на форматированное значение даты и времени **dtToConvert**. Список возможных заполнителей приведен в [Приложении А](#_Приложение_А._Заполнители). Все остальные символы строки **sFormatString** останутся без изменений. Если размер результирующей строки превышает 255 символов, то она будет обрезана до 255 символов.

**Таблица 2.15 – Описание входов и выходов функции DT\_TO\_STRING\_FORMAT**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| dtToConvert | DT | Метка времени |
| sFormatString | STRING(255) | Форматированная строка |
| **Выходные переменные** | | |
| DT\_TO\_STRING\_FORMAT | STRING(255) | Форматированная строка с меткой времени |

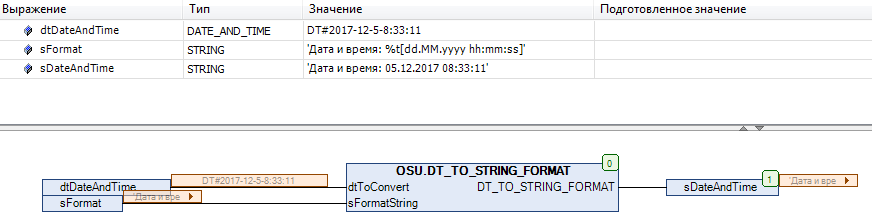


Рисунок 2.18 – Пример использования функции DT\_TO\_STRING\_FORMAT на языке CFC

### Функция DATE\_TO\_STRING\_FORMAT

Функция **DATE\_TO\_STRING\_FORMAT** заменяет в строке **sFormatString** первое вхождение подстроки типа %t[<заполнители>] на форматированное значение даты **dToConvert**. Список возможных заполнителей приведен в [Приложении А](#_Приложение_А._Заполнители). Все остальные символы строки **sFormatString** останутся без изменений. Если размер результирующей строки превышает 255 символов, то она будет обрезана до 255 символов.

**Таблица 2.16 – Описание входов и выходов функции DATE\_TO\_STRING\_FORMAT**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| dToConvert | DATE | Дата |
| sFormatString | STRING(255) | Форматированная строка |
| **Выходные переменные** | | |
| DATE\_TO\_STRING\_FORMAT | STRING(255) | Форматированная строка с датой |

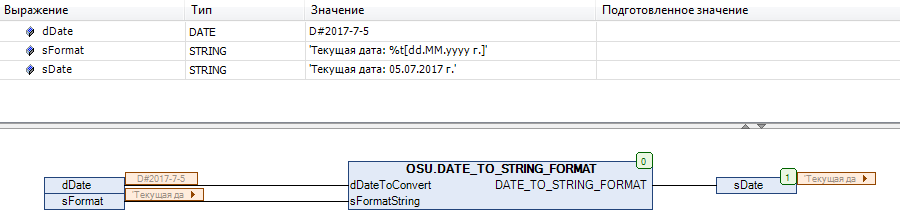


Рисунок 2.19 – Пример использования функции DATE\_TO\_STRING\_FORMAT на языке CFC

### Функция TOD\_TO\_STRING\_FORMAT

Функция **TOD\_TO\_STRING\_FORMAT** заменяет в строке **sFormatString** первое вхождение подстроки типа %t[<заполнители>] на форматированное значение времени суток **todToConvert**. Список возможных заполнителей приведен в [Приложении А](#_Приложение_А._Заполнители). Все остальные символы строки **sFormatString** останутся без изменений. Если размер результирующей строки превышает 255 символов, то она будет обрезана до 255 символов.

**Таблица 2.17 – Описание входов и выходов функции TOD\_TO\_STRING\_FORMAT**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| todToConvert | TOD | Время суток |
| sFormatString | STRING(255) | Форматированная строка |
| **Выходные переменные** | | |
| TOD\_TO\_STRING\_FORMAT | STRING(255) | Форматированная строка с датой |

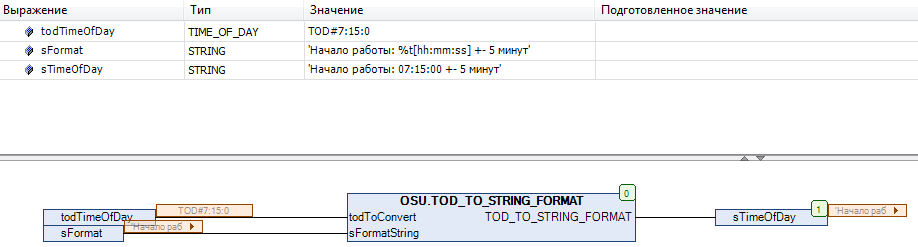


Рисунок 2.20 – Пример использования функции TOD\_TO\_STRING\_FORMAT на языке CFC

### Функция FindSubstringPosAfterN

Функция **FindSubstringPosAfterN** возвращает позицию первого вхождения искомой подстроки **sWhatToFind** в исходную строку **sSource**. Начальная позиция для поиска определяется входом **uiSearhFrom**. Если искомая подстрока не найдена, то функция возвращает 0. Строковые переменные функции имеют тип **STRING**.

**Таблица 2.18 – Описание входов и выходов функции FindSubstringPosAfterN**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sSource | STRING(255) | Исходная строка |
| sWhatToFind | STRING(255) | Искомая подстрока |
| uiSearchFrom | UINT | Начальная позиция для поиска |
| **Выходные переменные** | | |
| FindSubstringPosAfterN | UINT | Позиция вхождения искомой подстроки в исходную строку |

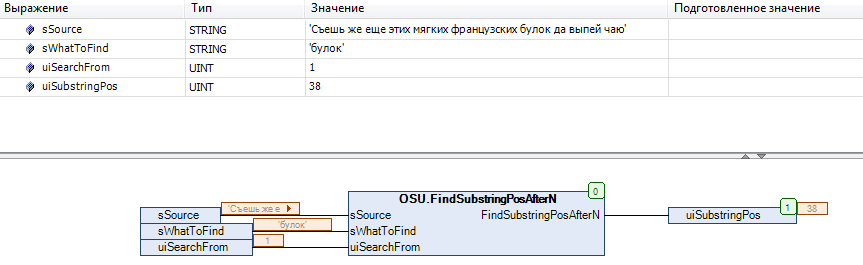


Рисунок 2.21 – Пример использования функции FindSubstringPosAfterN на языке CFC

### Функция WFindSubstringPosAfterN

Функция **WFindSubstringPosAfterN** возвращает позицию первого вхождения искомой подстроки **wsWhatToFind** в исходную строку **wsSource**. Начальная позиция для поиска определяется входом **uiSearhFrom**. Если искомая подстрока не найдена, то функция возвращает 0. Строковые переменные функции имеют тип **WSTRING**.

**Таблица 2.19 – Описание входов и выходов функции WFindSubstringPosAfterN**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsSource | WSTRING(255) | Исходная строка |
| wsWhatToFind | WSTRING(255) | Искомая подстрока |
| uiSearchFrom | UINT | Начальная позиция для поиска |
| **Выходные переменные** | | |
| WFindSubstringPosAfterN | UINT | Позиция вхождения искомой подстроки в исходную строку |

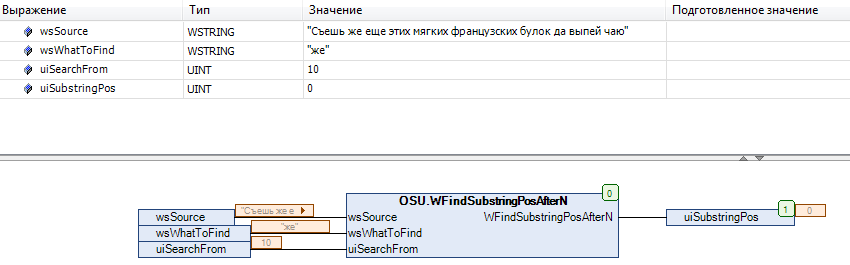


Рисунок 2.22 – Пример использования функции WFindSubstringPosAfterN на языке CFC

### Функция ReplaceSubstring

Функция **ReplaceSubstring** заменяет первое вхождение искомой подстроки **sWhatToReplace** в исходной строке **sSource** на подстроку **sReplaceWith**. Если искомая подстрока не найдена, то функция возвращает исходную строку. Все переменные функции имеют тип **STRING**.

**Таблица 2.20 – Описание входов и выходов функции ReplaceSubstring**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sSource | STRING(255) | Исходная строка |
| sWhatToReplace | STRING(255) | Искомая подстрока |
| sReplaceWith | STRING(255) | Замещающая подстрока |
| **Выходные переменные** | | |
| ReplaceSubstring | STRING(255) | Строка с замещенной подстрокой |

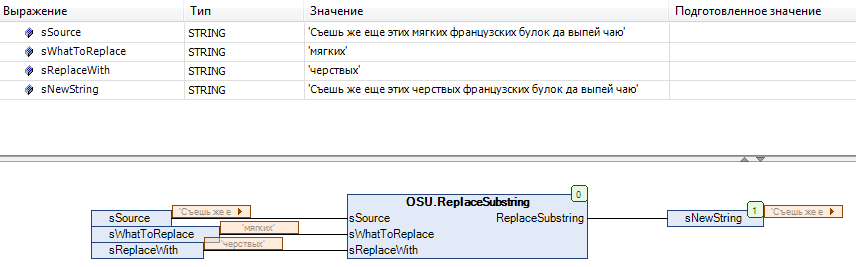


Рисунок 2.23 – Пример использования функции ReplaceSubstring на языке CFC

### Функция WReplaceSubstring

Функция **WReplaceSubstring** заменяет первое вхождение искомой подстроки **wsWhatToReplace** в исходной строке **wsSource** на подстроку **wsReplaceWith**. Если искомая подстрока не найдена, то функция возвращает исходную строку. Все переменные функции имеют тип **WSTRING**.

**Таблица 2.21 – Описание входов и выходов функции WReplaceSubstring**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsSource | WSTRING(255) | Исходная строка |
| wsWhatToReplace | WSTRING(255) | Искомая подстрока |
| wsReplaceWith | WSTRING(255) | Замещающая подстрока |
| **Выходные переменные** | | |
| WReplaceSubstring | WSTRING(255) | Строка с замещенной подстрокой |

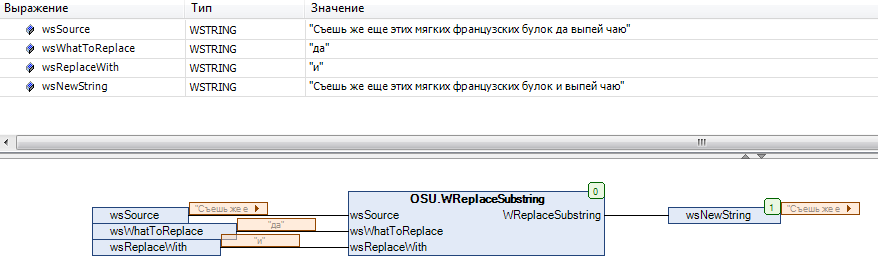


Рисунок 2.24 – Пример использования функции WReplaceSubstring на языке CFC

### Функция ReplaceAllSubstrings

Функция **ReplaceAllSubstrings** заменяет все вхождения искомой подстроки **sWhatToReplace** в исходной строке **sSource** на подстроку **sReplaceWith**. Если искомая подстрока не найдена, то функция возвращает исходную строку. Все переменные функции имеют тип **STRING**.

**Таблица 2.22 – Описание входов и выходов функции ReplaceAllSubstrings**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sSource | STRING(255) | Исходная строка |
| sWhatToReplace | STRING(255) | Искомая подстрока |
| sReplaceWith | STRING(255) | Замещающая подстрока |
| **Выходные переменные** | | |
| ReplaceAllSubstrings | STRING(255) | Строка с замещенными подстроками |

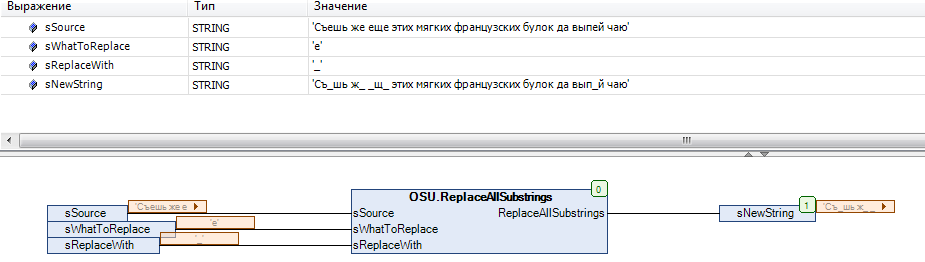


Рисунок 2.25 – Пример использования функции ReplaceAllSubstrings на языке CFC

### Функция WReplaceAllSubstrings

Функция **WReplaceAllSubstrings** заменяет все вхождения искомой подстроки **wsWhatToReplace** в исходной строке **wsSource** на подстроку **wsReplaceWith**. Если искомая подстрока не найдена, то функция возвращает исходную строку. Все переменные функции имеют тип **WSTRING**.

**Таблица 2.23 – Описание входов и выходов функции WReplaceAllSubstrings**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsSource | WSTRING(255) | Исходная строка |
| wsWhatToReplace | WSTRING(255) | Искомая подстрока |
| wsReplaceWith | WSTRING(255) | Замещающая подстрока |
| **Выходные переменные** | | |
| WReplaceAllSubstrings | WSTRING(255) | Строка с замещенными подстроками |

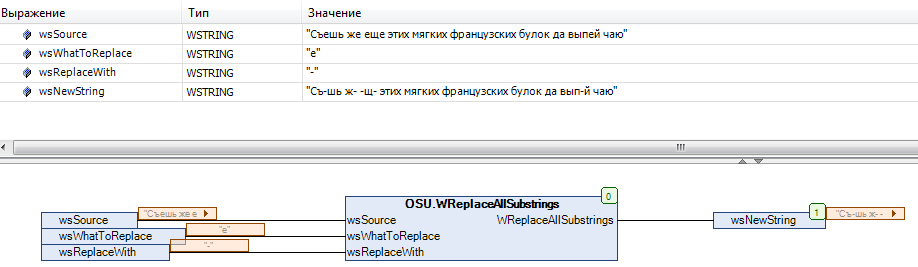


Рисунок 2.26 – Пример использования функции WReplaceAllSubstrings на языке CFC

### Функция CONCAT4

Функция **CONCAT4** объединяет входные строки типа STRING **sInputString1…4** в одну строку.

**Таблица 2.24 – Описание входов и выходов функции CONCAT4**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sInputString1…4 | STRING(255) | Исходные строки |
| **Выходные переменные** | | |
| CONCAT4 | STRING(255) | Объединенная строка |

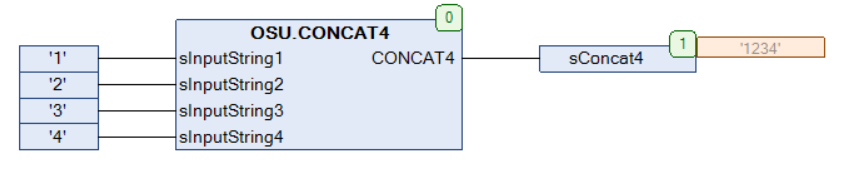


Рисунок 2.27 – Пример использования функции CONCAT4 на языке CFC

### Функция WCONCAT4

Функция **WCONCAT4** объединяет входные строки типа WSTRING **wsInputString1…4** в одну строку.

**Таблица 2.25 – Описание входов и выходов функции WCONCAT4**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsInputString1…4 | WSTRING(255) | Исходные строки |
| **Выходные переменные** | | |
| WCONCAT4 | WSTRING(255) | Объединенная строка |

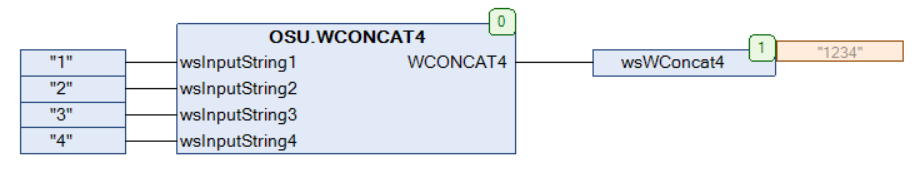


Рисунок 2.28 – Пример использования функции WCONCAT4 на языке CFC

### Функция CONCAT8

Функция **CONCAT8** объединяет входные строки типа STRING **sInputString1…8** в одну строку.

**Таблица 2.26 – Описание входов и выходов функции CONCAT8**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sInputString1…8 | STRING(255) | Исходные строки |
| **Выходные переменные** | | |
| CONCAT8 | STRING(255) | Объединенная строка |

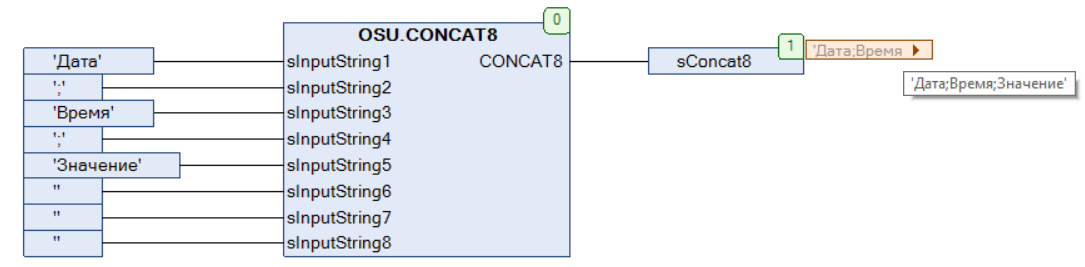


Рисунок 2.29 – Пример использования функции CONCAT8 на языке CFC

### Функция WCONCAT8

Функция **WCONCAT8** объединяет входные строки типа WSTRING **wsInputString1…8** в одну строку.

**Таблица 2.27 – Описание входов и выходов функции WCONCAT8**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsInputString1…8 | WSTRING(255) | Исходные строки |
| **Выходные переменные** | | |
| WCONCAT8 | WSTRING(255) | Объединенная строка |

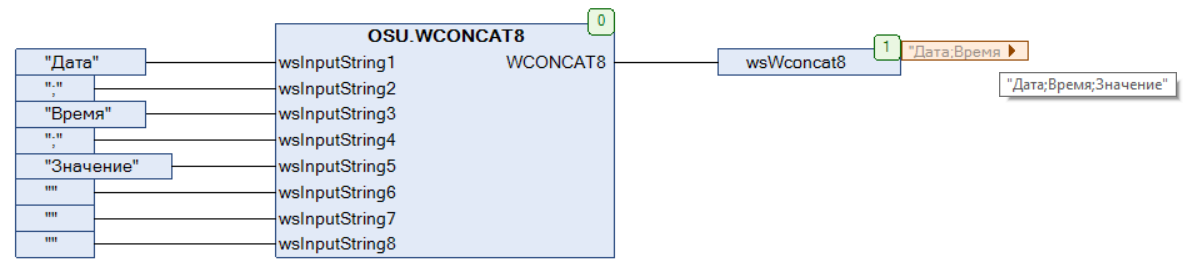


Рисунок 2.30 – Пример использования функции WCONCAT8 на языке CFC

### Функция ADD\_CHAR

Функция **ADD\_CHAR** дополняет строку типа STRING **sInputString** символом **sAddChar** до длины **usiTargetLen** справа (при **xRight** := **TRUE**) или слева (при **xRight** := **FALSE**).

Если длина sInputString> usiTargetLen, то функция возвращает **sInputString** без преобразований.

**Таблица 2.28 – Описание входов и выходов функции ADD\_CHAR**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sInputString | STRING(255) | Исходная строка |
| usiTargetLen | USINT | Длина результирующей строки |
| sAddChar | STRING(1) | Символ-заполнитель |
| xRight | BOOL | Режим дополнения строки (**TRUE** – справа, **FALSE** – слева) |
| **Выходные переменные** | | |
| ADD\_CHAR | STRING(255) | Строка, дополненная символами-заполнителями |

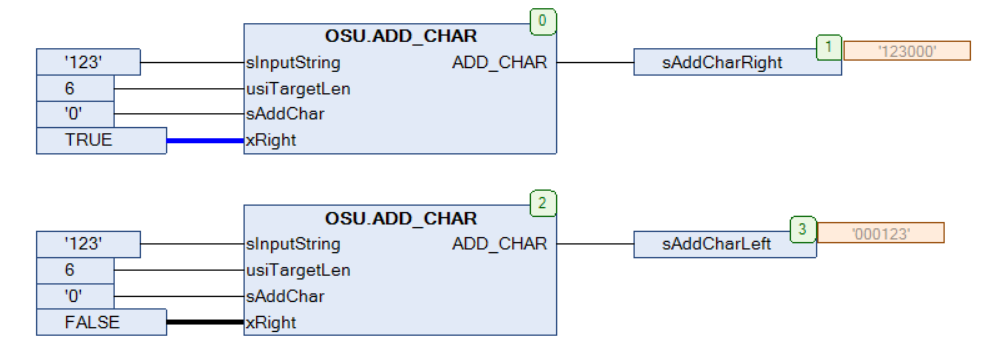


Рисунок 2.31 – Пример использования функции ADD\_CHAR на языке CFC

### Функция WADD\_CHAR

Функция **WADD\_CHAR** дополняет строку типа WSTRING **sInputString** символом **wsAddChar** до длины **usiTargetLen** справа (при **xRight** := **TRUE**) или слева (при **xRight** := **FALSE**).

Если длина wsInputString> usiTargetLen, то функция возвращает **wsInputString** без преобразований.

**Таблица 2.29 – Описание входов и выходов функции WADD\_CHAR**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wsInputString | WSTRING(255) | Исходная строка |
| usiTargetLen | USINT | Длина результирующей строки |
| wsAddChar | WSTRING(1) | Символ-заполнитель |
| xRight | BOOL | Режим дополнения строки (**TRUE** – справа, **FALSE** – слева) |
| **Выходные переменные** | | |
| WADD\_CHAR | WSTRING(255) | Строка, дополненная символами-заполнителями |

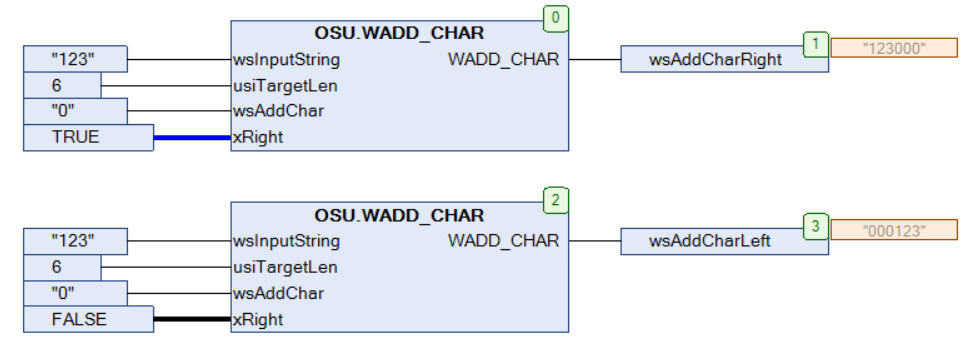


Рисунок 2.32 – Пример использования функции WADD\_CHAR на языке CFC

### Функция HEX\_STR\_TO\_WORD

Функция **HEX\_STR\_TO\_WORD** конвертирует строку с HEX-значением **sInputString** и префиксом **sPrefix** в переменную типа **WORD**, содержащую это значение в целочисленном виде. Исходная строка **sInputString** может включать в себя до 4 символов префикса и до 4 символов значения (от 0 до FFFF). Регистр HEX-символов не учитывается.

Если в исходной строке префикс отсутствует, то входу **sPrefix** должно быть присвоено значение ‘’.

**Таблица 2.30 – Описание входов и выходов функции HEX\_STR\_TO\_WORD**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sInputString | STRING(8) | Строка с HEX-значением |
| sPrefix | STRING(4) | Префикс исходной строки |
| **Выходные переменные** | | |
| HEX\_STR\_TO\_WORD | WORD | Результат преобразования |

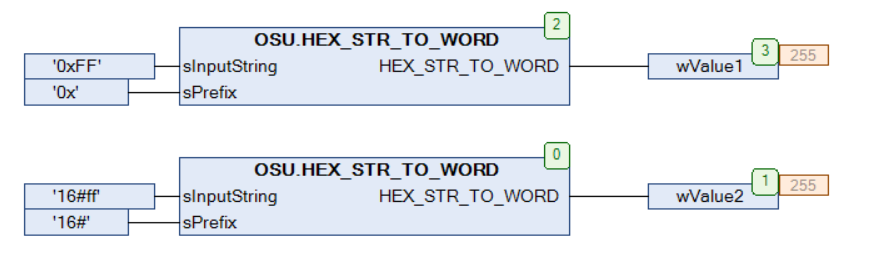


Рисунок 2.33 – Пример использования функции HEX\_STR\_TO\_WORD на языке CFC

### Функция WORD\_TO\_HEX\_STR

Функция **WORD\_TO\_HEX\_STR** конвертирует целочисленное значение типа WORD **wInput** в строку с его HEX-представлением и префиксом **sPrefix**.Если вход **xUpperCase** имеет значение **TRUE**, то HEX-символы строки имеют верхний регистр, если **FALSE** – то нижний.

**Таблица 2.31 – Описание входов и выходов функции WORD\_TO\_HEX\_STR**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| wInput | WORD | Исходное значение |
| xUpperCase | BOOL | Регистр HEX-символов (**TRUE** – верхний, **FALSE** – нижний) |
| sPrefix | STRING(4) | Префикс формируемой строки |
| **Выходные переменные** | | |
| WORD\_TO\_HEX\_STR | STRING(8) | Строка с префиксом и HEX-значением |

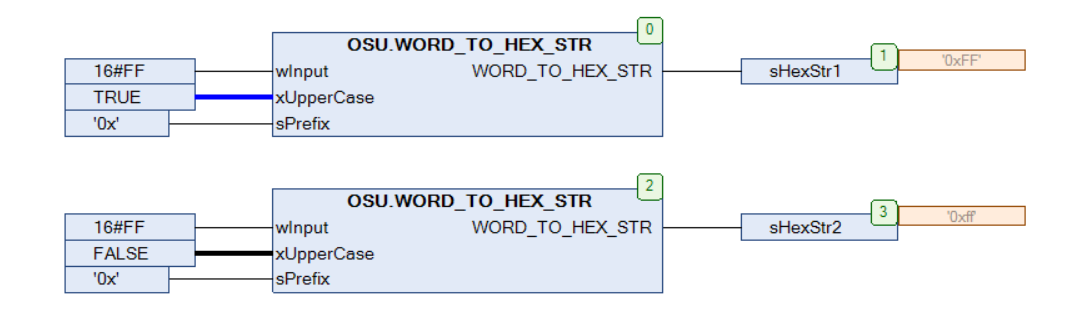


Рисунок 2.34 – Пример использования функции WORD\_TO\_HEX\_STR на языке CFC

### Функция BYTES\_TO\_IPSTRING

Функция **BYTES\_TO\_IPSTRING** возвращает строковое представление IP-адреса, заданного в виде массива байт **abyIpAddr**. Функция не работает в режиме эмуляции.

**Таблица 2.32 – Описание входов и выходов функции BYTES\_TO\_IPSTRING**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| abyIpAddr | ARRAY [0..3] OF BYTE | IP-адрес в виде массива байт |
| **Выходные переменные** | | |
| BYTES\_TO\_IPSTRING | STRING(15) | IP-адрес в строковом виде |

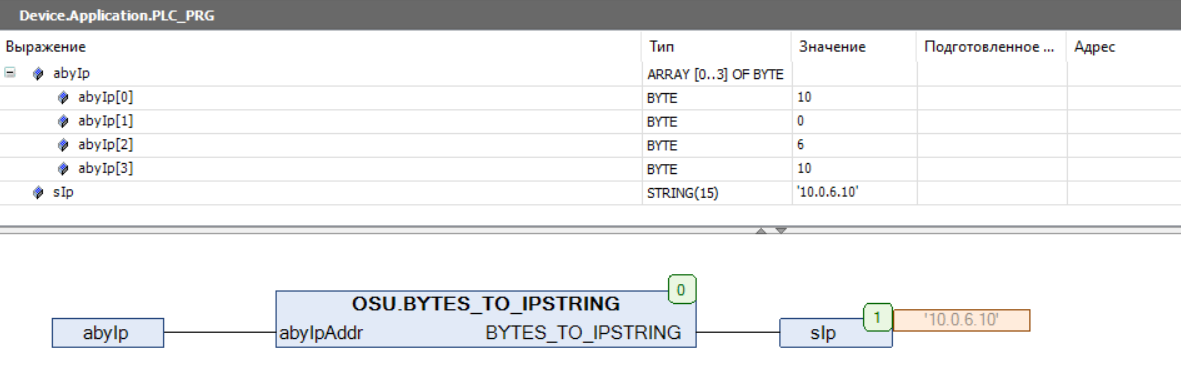


Рисунок 2.35 – Пример использования функции BYTES\_TO\_IPSTRING на языке CFC

### Функция IPSTRING\_TO\_BYTES

Функция **IPSTRING\_TO\_BYTES** конвертирует строковое представление IP-адреса **sIpAddr** в массив байт. Функция не работает в режиме эмуляции.

**Таблица 2.33 – Описание входов и выходов функции IPSTRING\_TO\_BYTES**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sIpAddr | STRING(15) | IP-адрес в строковом виде |
| **Выходные переменные** | | |
| IPSTRING\_TO\_BYTES | ARRAY [0..3] OF BYTE | IP-адрес в виде массива байт |

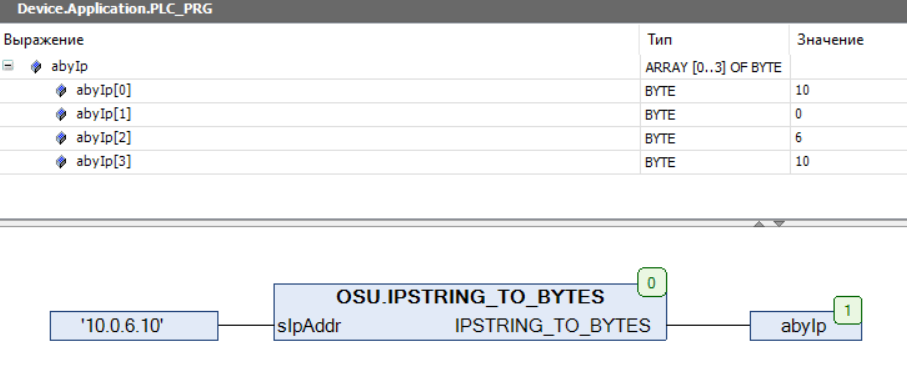


Рисунок 2.36 – Пример использования функции IPSTRING\_TO\_BYTES на языке CFC

### Функция UDINT\_TO\_IPSTRING

Функция **UDINT\_TO\_IPSTRING** возвращает строковое представление IP-адреса, заданного в виде переменной типа UDINT **udiIpAddr**.

**Таблица 2.34 – Описание входов и выходов функции UDINT\_TO\_IPSTRING**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| udiIpAddr | UDINT | IP-адрес в бинарном виде |
| **Выходные переменные** | | |
| UDINT\_TO\_IPSTRING | STRING | IP-адрес в строковом виде |

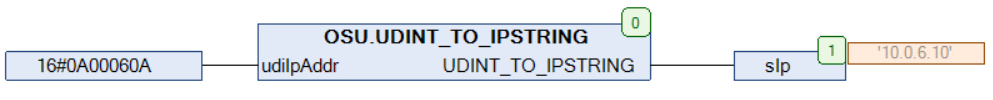


Рисунок 2.37 – Пример использования функции UDINT\_TO\_IPSTRING на языке CFC

### Функция IPSTRING\_TO\_UDINT

Функция **IPSTRING\_TO\_UDINT** конвертирует строковое представление IP-адреса **sIpAddr** в бинарный вид.

**Таблица 2.35 – Описание входов и выходов функции IPSTRING\_TO\_UDINT**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| sIpAddr | STRING | IP-адрес в строковом виде |
| **Выходные переменные** | | |
| IPSTRING\_TO\_UDINT | UDINT | IP-адрес в бинарном виде |

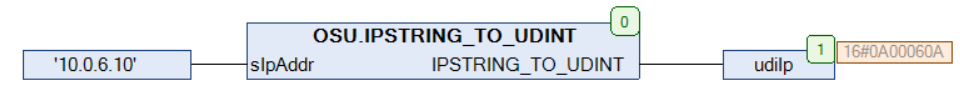


Рисунок 2.38 – Пример использования функции IPSTRING\_TO\_UDINT на языке CFC

### Функция MAC\_TO\_STRING

Функция **MAC\_TO\_STRING** возвращает строковое представление MAC-адреса, заданного в виде массива байт **abyMacAddr**.

**Таблица 2.36 – Описание входов и выходов функции MAC\_TO\_STRING**

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Тип** | **Описание** |
| **Входные переменные** | | |
| abyMacAddr | ARRAY [0..5] OF BYTE | MAC-адрес в виде массива байт |
| **Выходные переменные** | | |
| MAC\_TO\_STRING | STRING(17) | MAC-адрес в строковом виде |

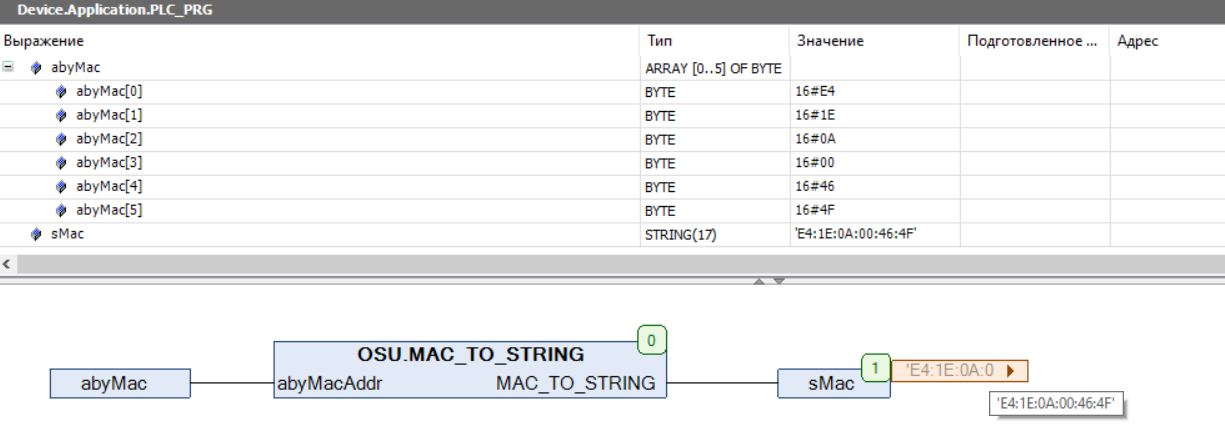


Рисунок 2.39 – Пример использования функции MAC\_TO\_STRING на языке CFC

# Приложение А. Заполнители формата времени

|  |  |  |  |
| --- | --- | --- | --- |
| **Заполнитель** | **Описание** | **Пример отображения** | **Используется в функциях** |
| d | День в виде числа (1–31) | 8 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT), [DATE\_TO\_STRING\_FORMAT](#_2.3.16_Функция_DATE_TO_STRING_FORMA) |
| dd | День с ведущим нулем (01–31) | 08 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT), [DATE\_TO\_STRING\_FORMAT](#_2.3.16_Функция_DATE_TO_STRING_FORMA) |
| M | Месяц в виде числа (1–12) | 8 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT), [DATE\_TO\_STRING\_FORMAT](#_2.3.16_Функция_DATE_TO_STRING_FORMA) |
| MM | Месяц с ведущим нулем (01–12) | 08 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT), [DATE\_TO\_STRING\_FORMAT](#_2.3.16_Функция_DATE_TO_STRING_FORMA) |
| y | Год века (0–99) | 8 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT), [DATE\_TO\_STRING\_FORMAT](#_2.3.16_Функция_DATE_TO_STRING_FORMA) |
| yy | Год века с ведущим нулем (00–99) | 08 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT), [DATE\_TO\_STRING\_FORMAT](#_2.3.16_Функция_DATE_TO_STRING_FORMA) |
| yyyy | Год | 2008 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT), [DATE\_TO\_STRING\_FORMAT](#_2.3.16_Функция_DATE_TO_STRING_FORMA) |
| HH | Час в 24-часовом формате (01–24) | 08 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT),  [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |
| hh | Час в 12-часовом формате (01–12) | 08 (и для 8-00, и для 20-00) | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT),  [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |
| m | Минуты (0 – 59) | 8 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT),  [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |
| mm | Минуты с ведущим нулем (00–59) | 08 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT),  [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |
| s | Секунды (0 – 59) | 8 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT),  [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |
| ss | Секунды с ведущим нулем (00–59) | 08 | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT),  [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |
| ms | Миллисекунды c ведущим нулем (000–999) | 008 | [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |
| t | Идентификатор для 12-часового формата: A (часы < 12) и P (часы > 12) | A (8 часов) | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT),  [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |
| tt | Идентификатор для 12-часового формата: AM (часы < 12) и PM (часы > 12) | PM (15 часов) | [DT\_TO\_STRING\_FORMAT](#_2.3.15_Функция_DT_TO_STRING_FORMAT),  [TOD\_TO\_STRING\_FORMAT](#_2.3.17_Функция_TOD_TO_STRING_FORMAT) |