



# **CODESYS V3.5**

**Описание библиотеки OwenStringUtils**



**Руководство пользователя**

01.12.2018

версия 2.0

## Оглавление

<b>1</b>	<b>Цель документа.....</b>	<b>3</b>
<b>2</b>	<b>Описание библиотеки OwenStringUtils .....</b>	<b>4</b>
2.1	Установка библиотеки.....	4
2.2	Добавление библиотеки в проект CODESYS.....	5
2.3	Описание библиотеки .....	6
2.3.1	Функция CP1251_TO_UNICODE .....	6
2.3.2	Функция UNICODE_TO_CP1251 .....	7
2.3.3	Функция Before .....	8
2.3.4	Функция WBefore .....	9
2.3.5	Функция After .....	10
2.3.6	Функция WAfter .....	11
2.3.7	Функция Between .....	12
2.3.8	Функция WBetween .....	13
2.3.9	Функция LowerCase.....	14
2.3.10	Функция WLowerCase .....	15
2.3.11	Функция UpperCase.....	16
2.3.12	Функция WUpperCase .....	17
2.3.13	Функция REAL_TO_STRING_FORMAT .....	18
2.3.14	Функция LREAL_TO_STRING_FORMAT .....	19
2.3.15	Функция DT_TO_STRING_FORMAT .....	20
2.3.16	Функция DATE_TO_STRING_FORMAT .....	21
2.3.17	Функция TOD_TO_STRING_FORMAT .....	22
2.3.18	Функция FindSubstringPosAfterN .....	23
2.3.19	Функция WFindSubstringPosAfterN.....	24
2.3.20	Функция ReplaceSubstring .....	25
2.3.21	Функция WReplaceSubstring .....	26
2.3.22	Функция ReplaceAllSubstrings.....	27
2.3.23	Функция WReplaceAllSubstrings .....	28
<b>3</b>	<b>Приложение А. Заполнители формата времени .....</b>	<b>29</b>

## 1 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 **3.5.4.7**.



### 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**.

## 2 Описание библиотеки OwenStringUtils

### 2.1 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**:

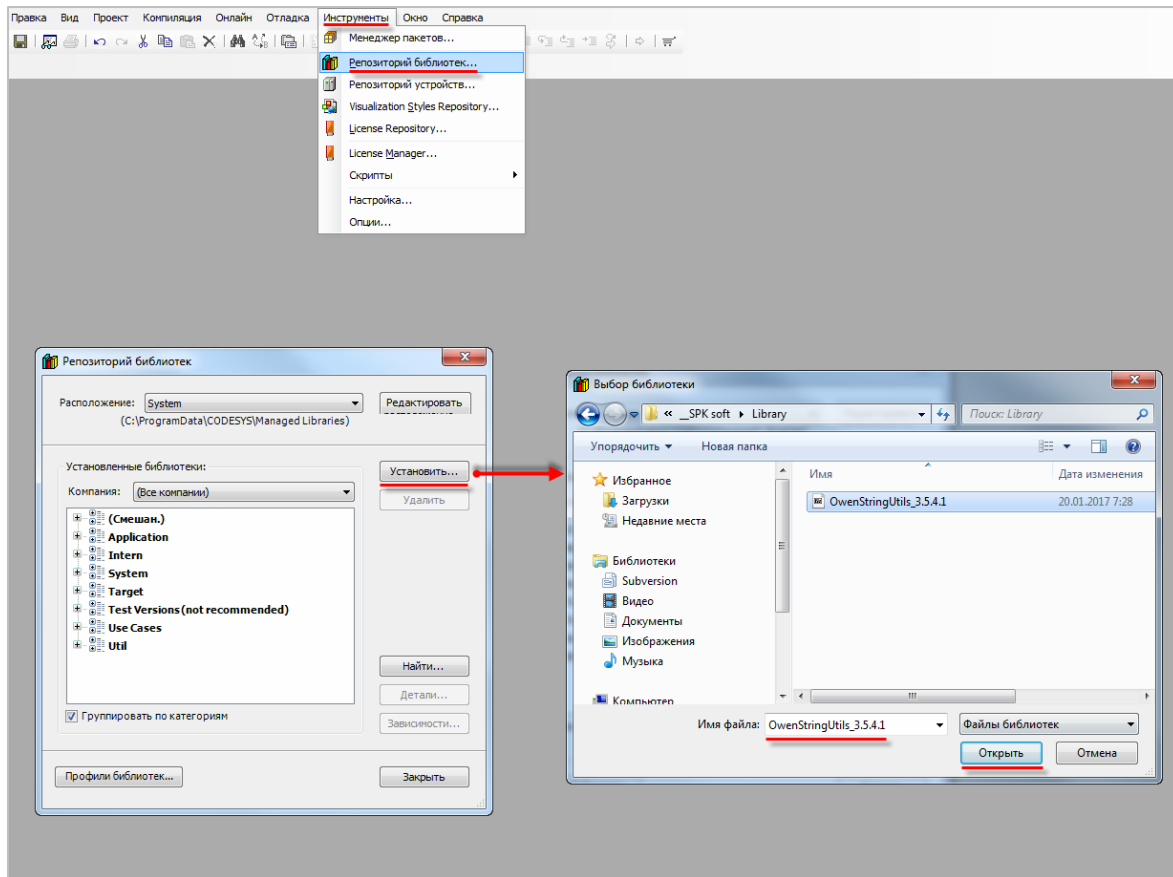


Рисунок 2.1 – Установка библиотеки в среду CODESYS

## 2.2 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**.

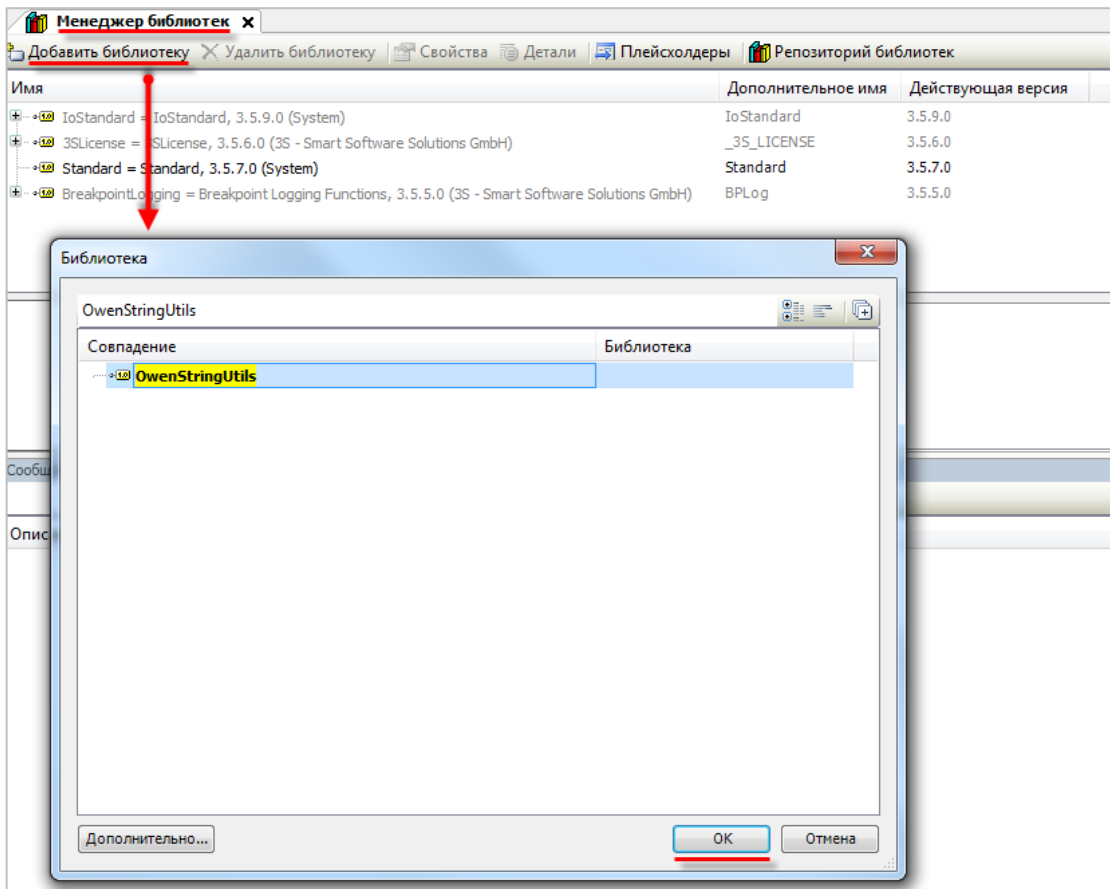


Рисунок 2.2 – Добавление библиотеки OwenStringUtils

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

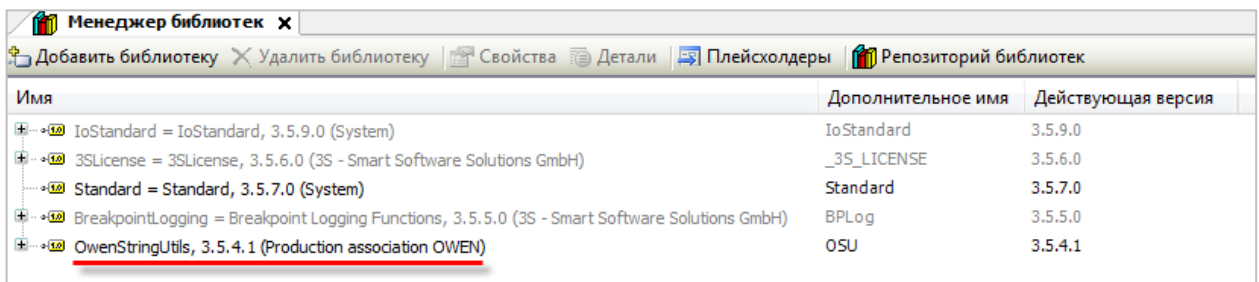


Рисунок 2.3 – Список библиотек проекта



### ПРИМЕЧАНИЕ

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

## 2.3 Описание библиотеки

### 2.3.1 Функция CP1251\_TO\_UNICODE

Функция **CP1251\_TO\_UNICODE** используется для конвертации переменной типа **STRING**, содержащей строку в кодировке [ASCII \(CP1251\)](#), в переменную типа **WSTRING**, содержащую строку в кодировке [Unicode \(UCS-2\)](#).

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

Имя переменной	Тип	Описание
<b>Входные переменные</b>		
sInputString	STRING	Исходная строка в кодировке <a href="#">ASCII (CP1251)</a>
<b>Выходные переменные</b>		
CP1251_TO_UNICODE	WSTRING	Строка в кодировке <a href="#">Unicode (UCS-2)</a>

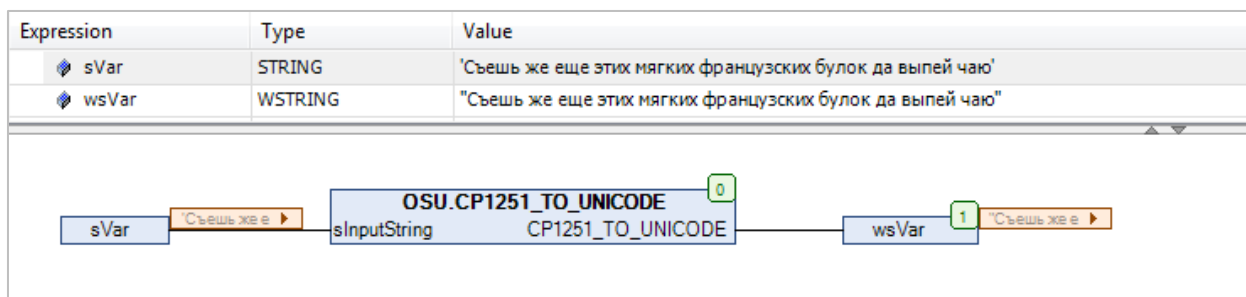


Рисунок 2.4 – Пример использования функции CP1251\_TO\_UNICODE на языке CFC

### 2.3.2 Функция UNICODE\_TO\_CP1251

Функция **UNICODE\_TO\_CP1251** используется для конвертации переменной типа **WSTRING**, содержащей строку в кодировке [Unicode \(UCS-2\)](#), в переменную типа **STRING**, содержащую строку в кодировке [ASCII \(CP1251\)](#)

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

Имя переменной	Тип	Описание
<b>Входные переменные</b>		
wsInputString	WSTRING	Исходная строка в кодировке <a href="#">Unicode (UCS-2)</a>
<b>Выходные переменные</b>		
UNICODE_TO_CP1251	STRING	Строка в кодировке <a href="#">ASCII (CP1251)</a>

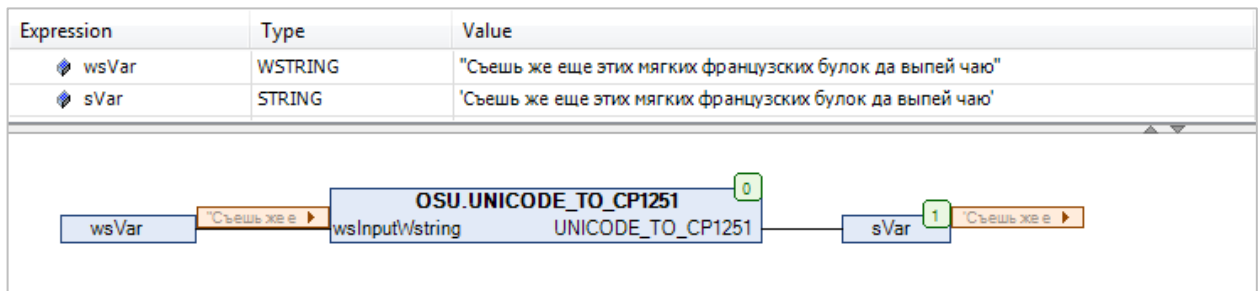


Рисунок 2.5 – Пример использования функции **UNICODE\_TO\_CP1251** на языке CFC

### 2.3.3 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
<b>Входные переменные</b>		
sSource	STRING	Исходная строка
sPostfix	STRING	Подстрока
<b>Выходные переменные</b>		
Before	STRING	Фрагмент исходной строки, предшествующий первому вхождению подстроки (не включая саму подстроку)

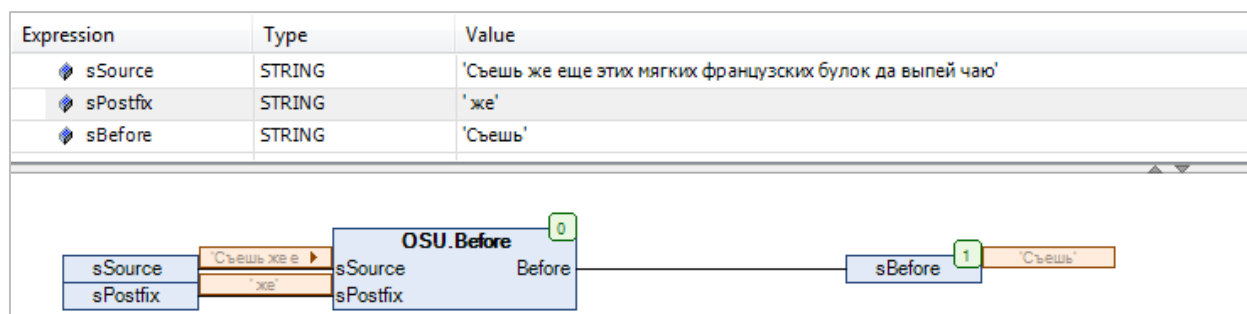


Рисунок 2.6 – Пример использования функции Before на языке CFC



### 2.3.4 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
<b>Входные переменные</b>		
wsSource	WSTRING	Исходная строка
wsPostfix	WSTRING	Подстрока
<b>Выходные переменные</b>		
WBefore	WSTRING	Фрагмент исходной строки, предшествующий первому вхождению подстроки (не включая саму подстроку)

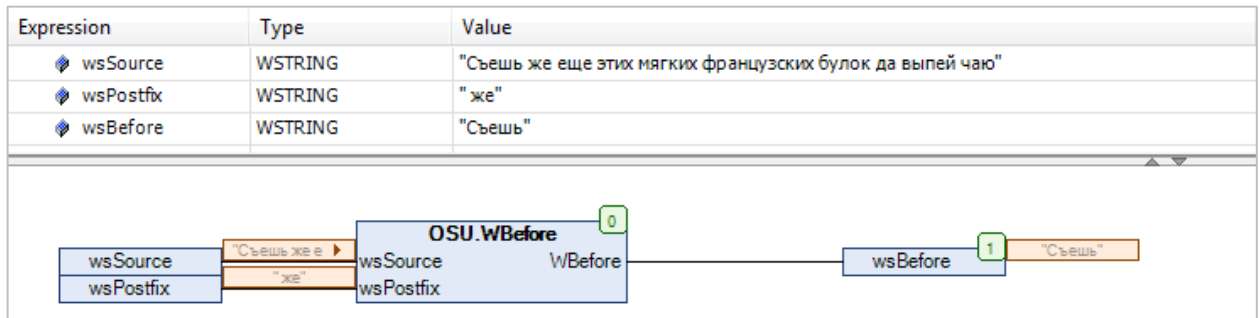


Рисунок 2.7 – Пример использования функции WBefore на языке CFC

### 2.3.5 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
<b>Входные переменные</b>		
sSource	STRING	Исходная строка
sPrefix	STRING	Подстрока
<b>Выходные переменные</b>		
After	STRING	Фрагмент исходной строки, следующий за первым вхождением подстроки (не включая саму подстроку)

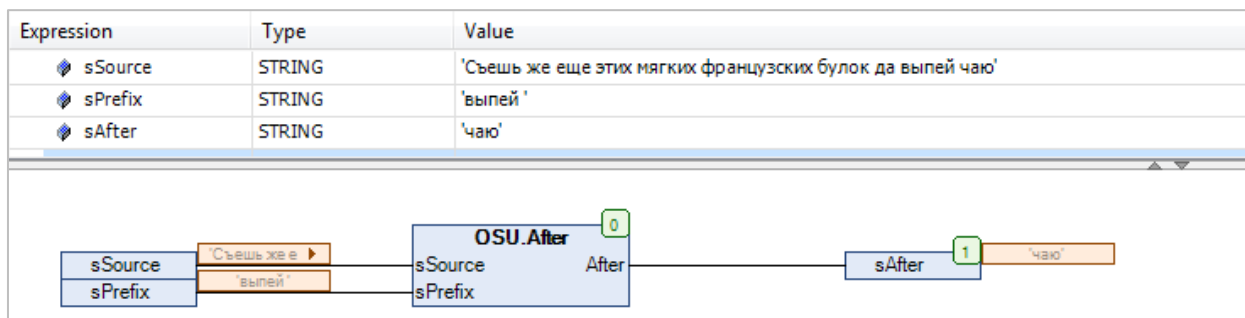


Рисунок 2.8 – Пример использования функции After на языке CFC

### 2.3.6 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
<b>Входные переменные</b>		
wsSource	WSTRING	Исходная строка
wsPrefix	WSTRING	Подстрока
<b>Выходные переменные</b>		
WAfter	WSTRING	Фрагмент исходной строки, следующий за первым вхождением подстроки (не включая саму подстроку)

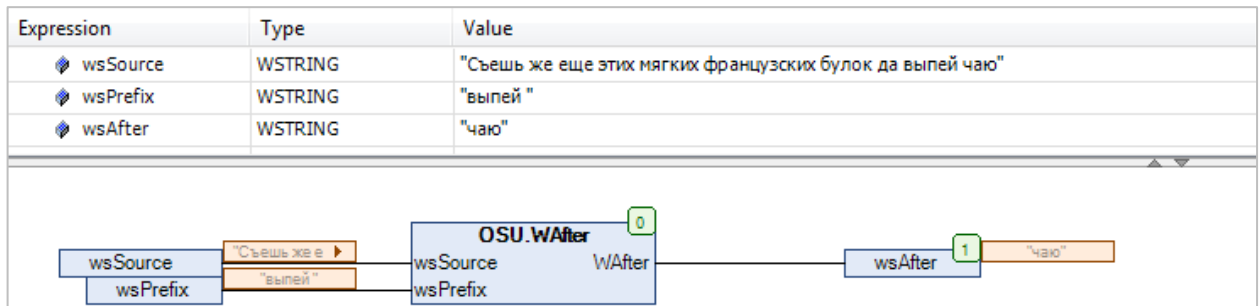


Рисунок 2.9 – Пример использования функции WAfter на языке CFC

### 2.3.7 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
<b>Входные переменные</b>		
sSource	STRING	Исходная строка
sPrefix	STRING	Начальная подстрока
sPostfix	STRING	Конечная подстрока
<b>Выходные переменные</b>		
Between	STRING	Фрагмент исходной строки, расположенный между первыми вхождениями начальной и конечной подстрок (не включая сами подстроки)

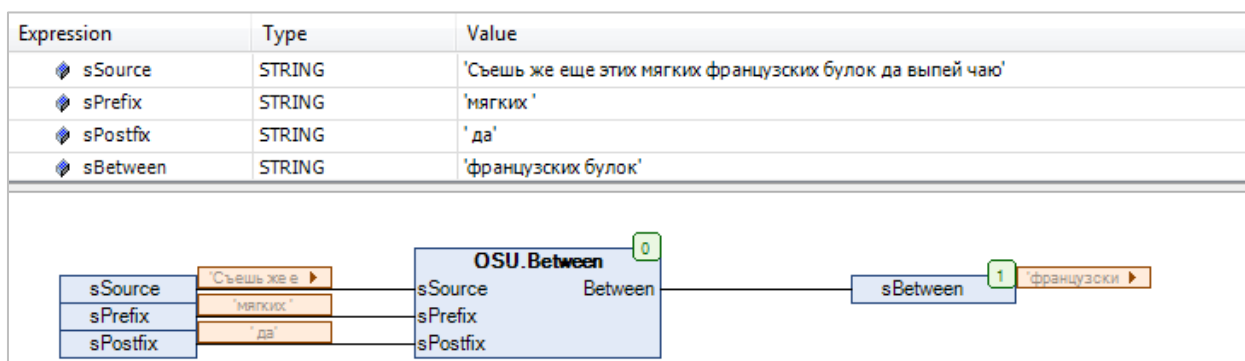


Рисунок 2.10 – Пример использования функции **Between** на языке CFC

### 2.3.8 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
<b>Входные переменные</b>		
wsSource	WSTRING	Исходная строка
wsPrefix	WSTRING	Начальная подстрока
wsPostfix	WSTRING	Конечная подстрока
<b>Выходные переменные</b>		
WBetween	WSTRING	Фрагмент исходной строки, расположенный между первыми вхождениями начальной и конечной подстрок (не включая сами подстроки)

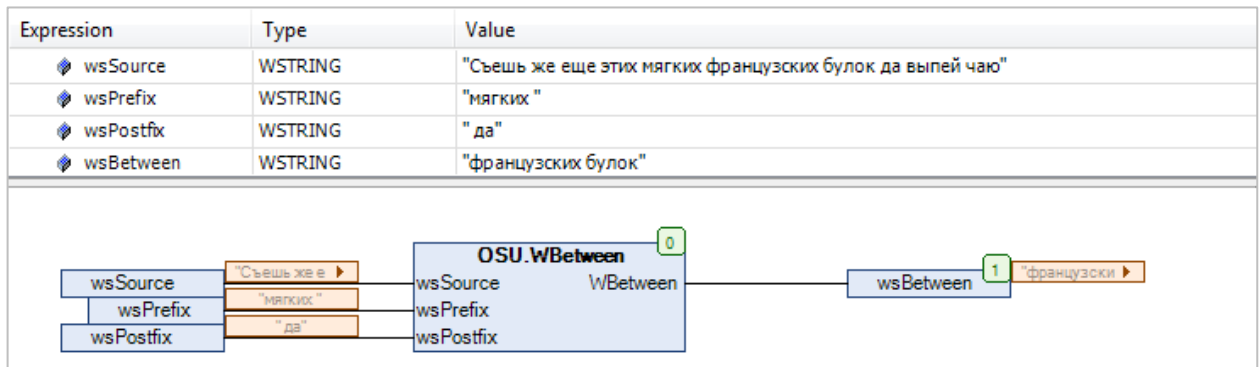


Рисунок 2.11 – Пример использования функции WBetween на языке CFC

### 2.3.9 Функция LowerCase

Функция **LowerCase** преобразует все символы исходной строки **sStringToConvert** (в кодировке [CP1251](#)) в нижний регистр. Все переменные функции имеют тип **STRING**.

Таблица 2.9 – Описание входов и выходов функции LowerCase

Имя переменной	Тип	Описание
<b>Входные переменные</b>		
sStringToConvert	STRING	Исходная строка
<b>Выходные переменные</b>		
LowerCase	STRING	Строка в нижнем регистре

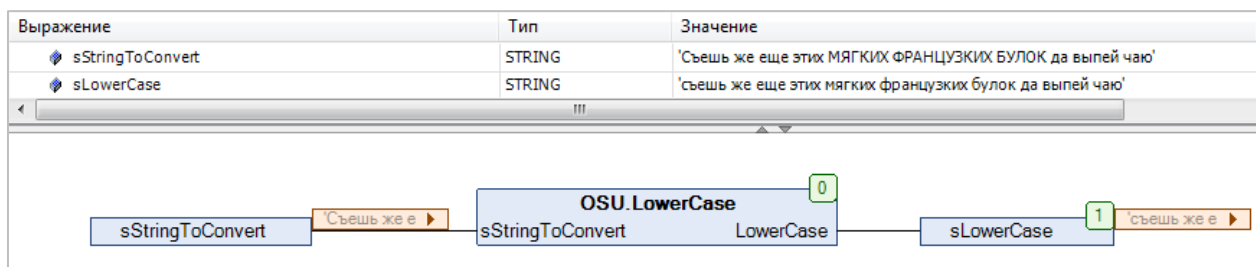


Рисунок 2.12 – Пример использования функции LowerCase на языке CFC

### 2.3.10 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
<b>Входные переменные</b>		
wsStringToConvert	WSTRING	Исходная строка
<b>Выходные переменные</b>		
WLowerCase	WSTRING	Строка в нижнем регистре

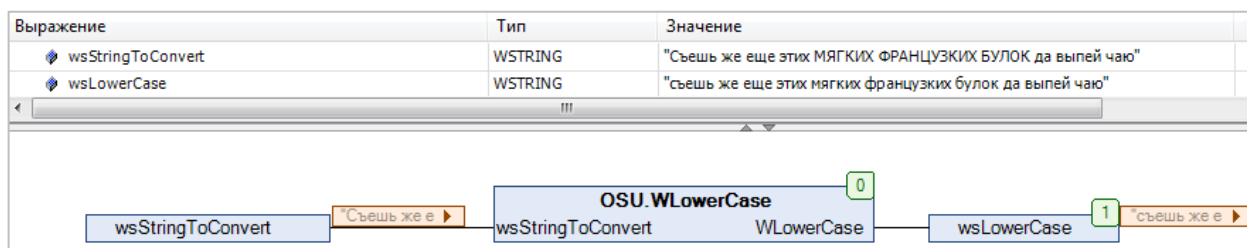


Рисунок 2.13 – Пример использования функции WLowerCase на языке CFC

### 2.3.11 Функция UpperCase

Функция **UpperCase** преобразует все символы исходной строки **sStringToConvert** (в кодировке [CP1251](#)) в верхний регистр. Все переменные функции имеют тип **STRING**.

Таблица 2.11 – Описание входов и выходов функции UpperCase

Имя переменной	Тип	Описание
<b>Входные переменные</b>		
sStringToConvert	STRING	Исходная строка
<b>Выходные переменные</b>		
UpperCase	STRING	Строка в верхнем регистре

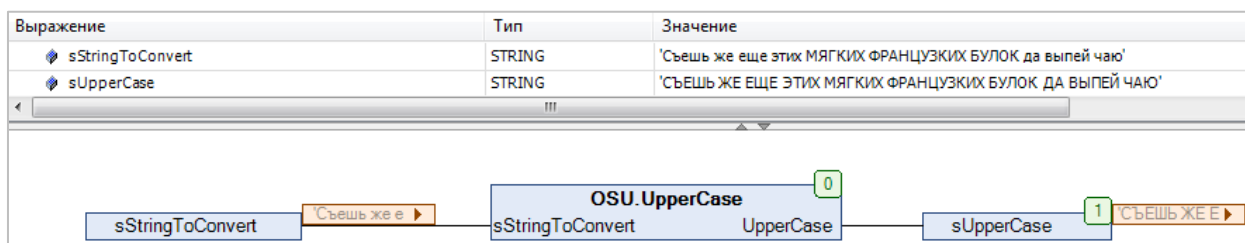


Рисунок 2.14 – Пример использования функции UpperCase на языке CFC



### 2.3.12 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
<b>Входные переменные</b>		
wsStringToConvert	WSTRING	Исходная строка
<b>Выходные переменные</b>		
WUpperCase	WSTRING	Строка в верхнем регистре

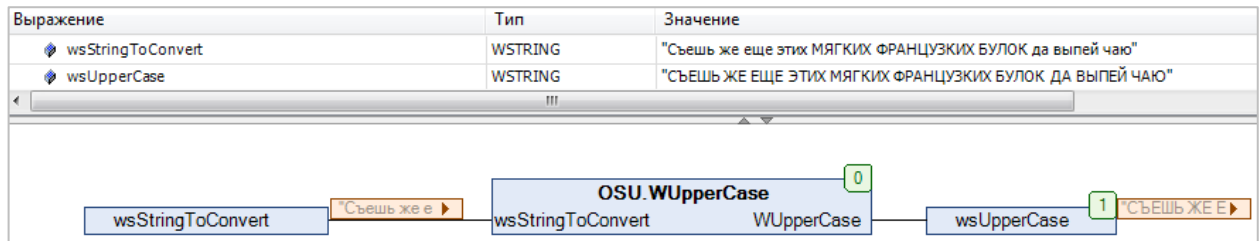


Рисунок 2.15 – Пример использования функции WUpperCase на языке CFC

### 2.3.13 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
<b>Входные переменные</b>		
rValueToConvert	REAL	Значение с плавающей точкой
usiSignificantDigitsCount	USINT	Количество знаков после разделителя
eDecimalSeparator	DECIMAL_SEPARATOR	Разделитель целой и дробной части
<b>Выходные переменные</b>		
REAL_TO_STRING_FORMAT	STRING(80)	Значение в виде форматированной строки

Выражение	Тип	Значение
rValue	REAL	11.2233
usiPrecision	USINT	3
eSeparator	DECIMAL_SEPARATOR	COMMA
sRealValue	STRING	'11,223'

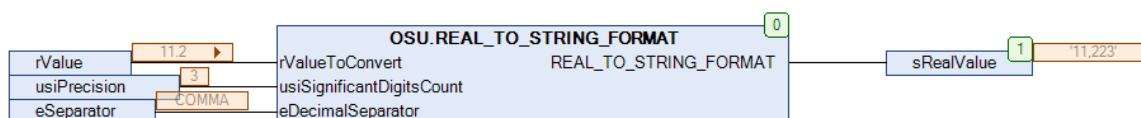


Рисунок 2.16 – Пример использования функции REAL\_TO\_STRING\_FORMAT на языке CFC

### 2.3.14 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
<b>Входные переменные</b>		
lrValueToConvert	LREAL	Значение с плавающей точкой
usiSignificantDigitsCount	USINT	Количество знаков после разделителя
eDecimalSeparator	DECIMAL_SEPARATOR	Разделитель целой и дробной части
<b>Выходные переменные</b>		
LREAL_TO_STRING_FORMAT	STRING(80)	Значение в виде форматированной строки

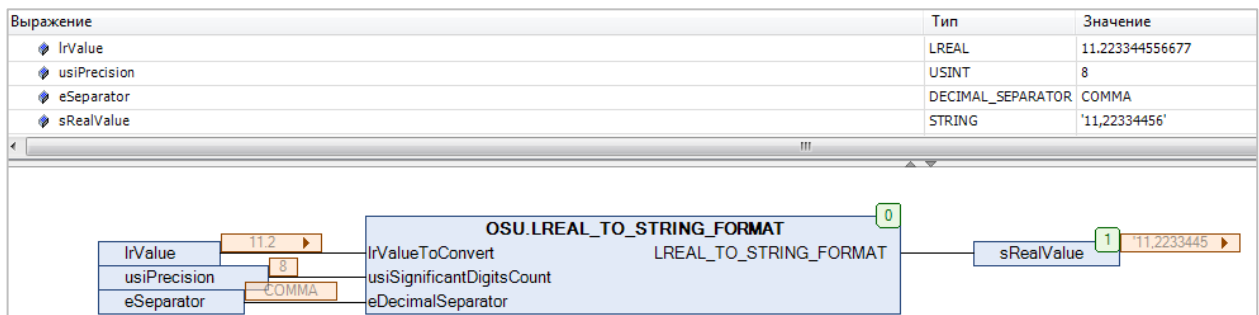


Рисунок 2.17 – Пример использования функции LREAL\_TO\_STRING\_FORMAT на языке CFC

### 2.3.15 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](#). 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
<b>Входные переменные</b>		
dtToConvert	DT	Метка времени
sFormatString	STRING	Форматированная строка
<b>Выходные переменные</b>		
DT_TO_STRING_FORMAT	STRING	Форматированная строка с меткой времени

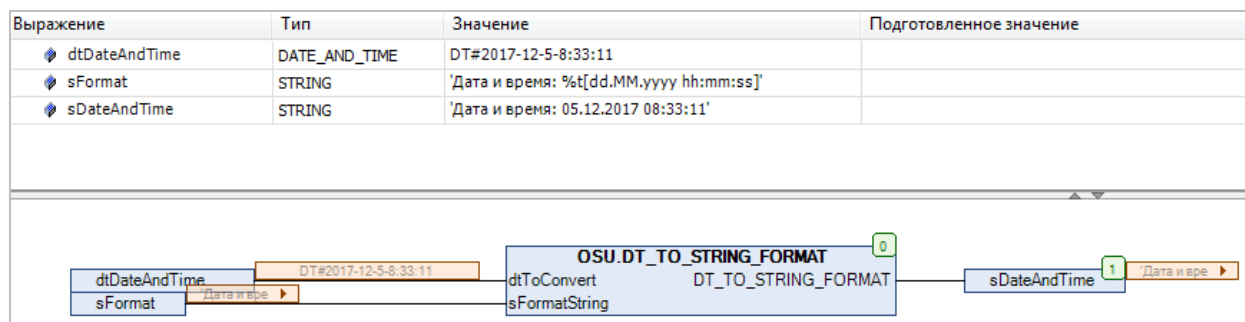


Рисунок 2.18 – Пример использования функции DT\_TO\_STRING\_FORMAT на языке CFC

### 2.3.16 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](#). 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
<b>Входные переменные</b>		
dToConvert	DATE	Дата
sFormatString	STRING	Форматированная строка
<b>Выходные переменные</b>		
DATE_TO_STRING_FORMAT	STRING	Форматированная строка с датой

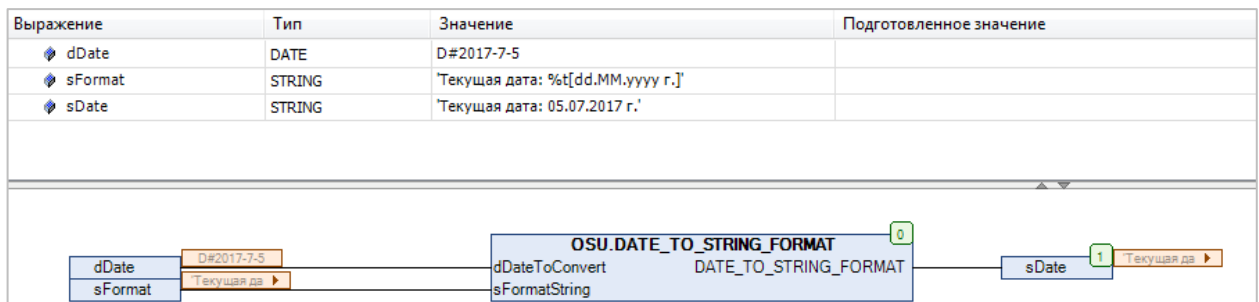


Рисунок 2.19 – Пример использования функции DATE\_TO\_STRING\_FORMAT на языке CFC

### 2.3.17 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](#). 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
<b>Входные переменные</b>		
todToConvert	TOD	Время суток
sFormatString	STRING	Форматированная строка
<b>Выходные переменные</b>		
TOD_TO_STRING_FORMAT	STRING	Форматированная строка с датой

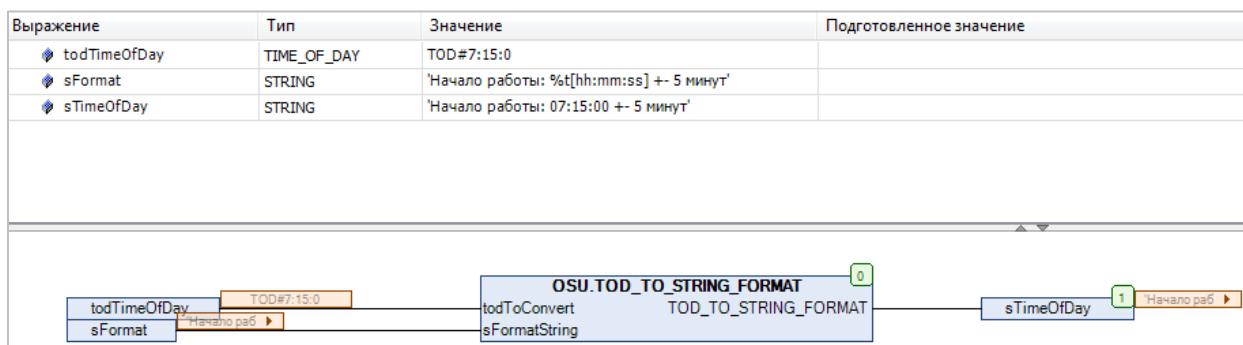


Рисунок 2.20 – Пример использования функции TOD\_TO\_STRING\_FORMAT на языке CFC

### 2.3.18 Функция FindSubstringPosAfterN

Функция **FindSubstringPosAfterN** возвращает позицию первого вхождения искомой подстроки **sWhatToFind** в исходную строку **sSource**. Начальная позиция для поиска определяется входом **uiSearchFrom**. 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
<b>Входные переменные</b>		
sSource	STRING	Исходная строка
sWhatToFind	STRING	Искомая подстрока
uiSearchFrom	UINT	Начальная позиция для поиска
<b>Выходные переменные</b>		
FindSubstringPosAfterN	UINT	Позиция вхождения искомой подстроки в исходную строку

Выражение	Тип	Значение	Подготовленное значение
sSource	STRING	'Съешь же еще этих мягких французских булок да выпей чаю'	
sWhatToFind	STRING	'булок'	
uiSearchFrom	UINT	1	
uiSubstringPos	UINT	38	

Рисунок 2.21 – Пример использования функции FindSubstringPosAfterN на языке CFC

### 2.3.19 Функция WFindSubstringPosAfterN

Функция **WFindSubstringPosAfterN** возвращает позицию первого вхождения искомой подстроки **wsWhatToFind** в исходную строку **wsSource**. Начальная позиция для поиска определяется входом **uiSearchFrom**. 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
<b>Входные переменные</b>		
wsSource	WSTRING	Исходная строка
wsWhatToFind	WSTRING	Искомая подстрока
uiSearchFrom	UINT	Начальная позиция для поиска
<b>Выходные переменные</b>		
WFindSubstringPosAfterN	UINT	Позиция вхождения искомой подстроки в исходную строку

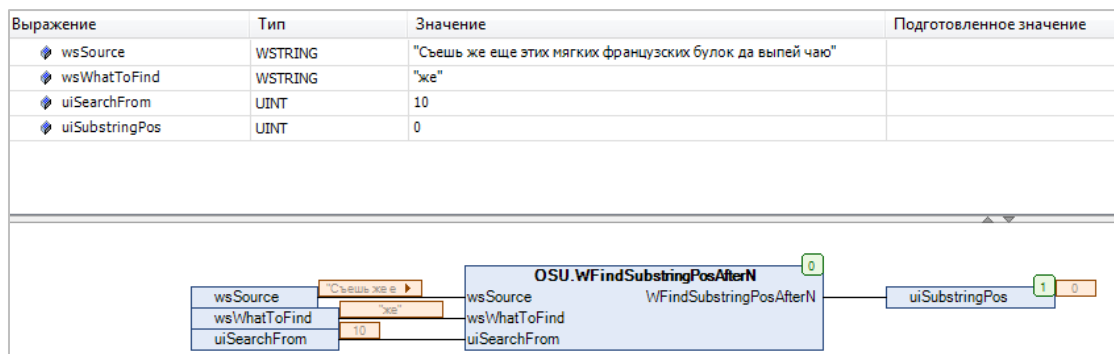


Рисунок 2.22 – Пример использования функции WFindSubstringPosAfterN на языке CFC



### 2.3.20 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
<b>Входные переменные</b>		
sSource	STRING	Исходная строка
sWhatToReplace	STRING	Искомая подстрока
sReplaceWith	STRING	Замещающая подстрока
<b>Выходные переменные</b>		
ReplaceSubstring	STRING	Строка с замещенной подстрокой

Выражение	Тип	Значение	Подготовленное значение
sSource	STRING	'Съешь же еще этих мягких французских булок да выпей чаю'	
sWhatToReplace	STRING	'мягких'	
sReplaceWith	STRING	'черствых'	
sNewString	STRING	'Съешь же еще этих черствых французских булок да выпей чаю'	

Рисунок 2.23 – Пример использования функции ReplaceSubstring на языке CFC

### 2.3.21 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
<b>Входные переменные</b>		
wsSource	WSTRING	Исходная строка
wsWhatToReplace	WSTRING	Искомая подстрока
wsReplaceWith	WSTRING	Замещающая подстрока
<b>Выходные переменные</b>		
WReplaceSubstring	WSTRING	Строка с замещенной подстрокой

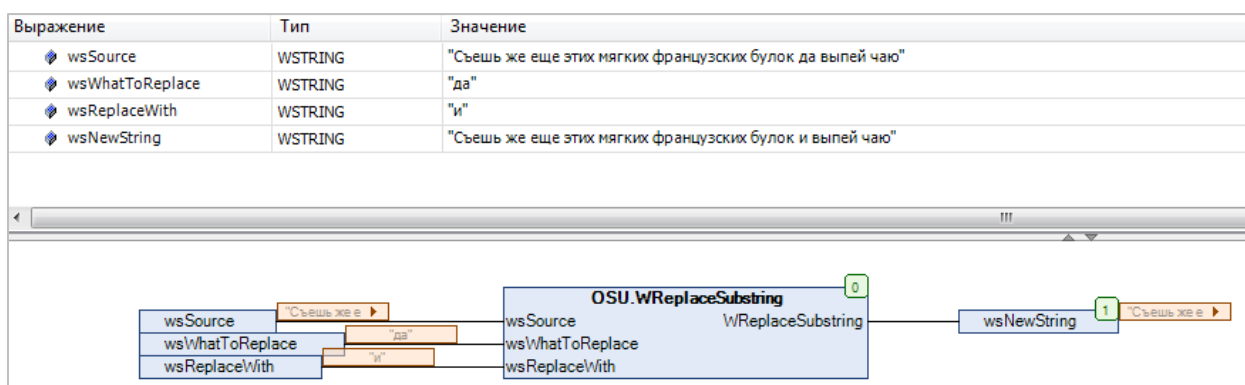


Рисунок 2.24 – Пример использования функции WReplaceSubstring на языке CFC

### 2.3.22 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
<b>Входные переменные</b>		
sSource	STRING	Исходная строка
sWhatToReplace	STRING	Искомая подстрока
sReplaceWith	STRING	Замещающая подстрока
<b>Выходные переменные</b>		
ReplaceAllSubstrings	STRING	Строка с замещенными подстроками

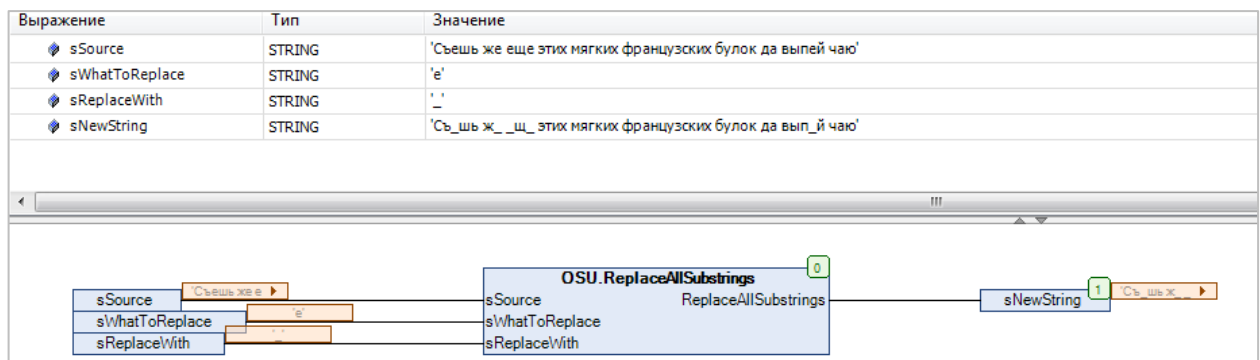


Рисунок 2.25 – Пример использования функции ReplaceAllSubstrings на языке CFC

### 2.3.23 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
<b>Входные переменные</b>		
wsSource	WSTRING	Исходная строка
wsWhatToReplace	WSTRING	Искомая подстрока
wsReplaceWith	WSTRING	Замещающая подстрока
<b>Выходные переменные</b>		
WReplaceAllSubstrings	WSTRING	Строка с замещенными подстроками

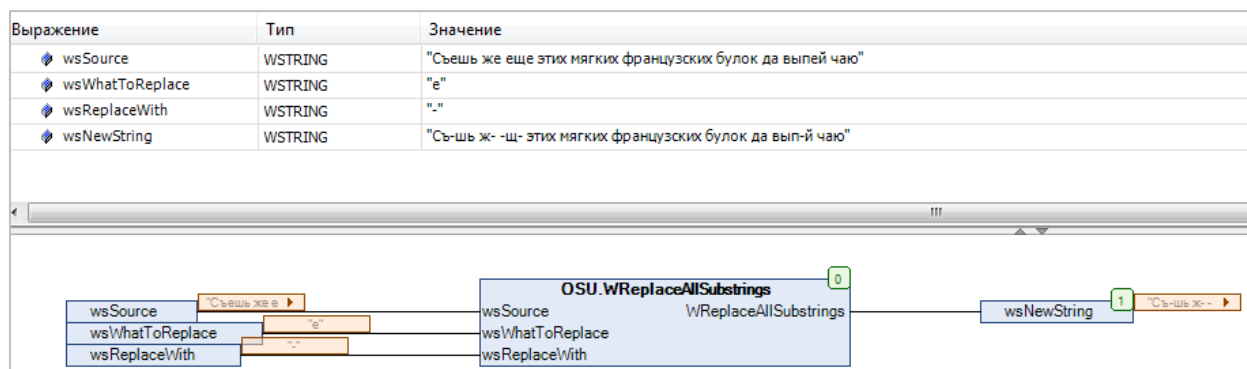


Рисунок 2.26 – Пример использования функции WReplaceAllSubstrings на языке CFC

### 3 Приложение А. Заполнители формата времени

Заполнитель	Описание	Пример отображения	Используется в функциях
d	День в виде числа (1–31)	8	<a href="#">DT TO STRING FORMAT</a> , <a href="#">DATE TO STRING FORMAT</a>
dd	День с ведущим нулем (01–31)	08	<a href="#">DT TO STRING FORMAT</a> , <a href="#">DATE TO STRING FORMAT</a>
M	Месяц в виде числа (1–12)	8	<a href="#">DT TO STRING FORMAT</a> , <a href="#">DATE TO STRING FORMAT</a>
MM	Месяц с ведущим нулем (01–12)	08	<a href="#">DT TO STRING FORMAT</a> , <a href="#">DATE TO STRING FORMAT</a>
y	Год века (0–99)	8	<a href="#">DT TO STRING FORMAT</a> , <a href="#">DATE TO STRING FORMAT</a>
yy	Год века с ведущим нулем (00–99)	08	<a href="#">DT TO STRING FORMAT</a> , <a href="#">DATE TO STRING FORMAT</a>
yyyy	Год	2008	<a href="#">DT TO STRING FORMAT</a> , <a href="#">DATE TO STRING FORMAT</a>
HH	Час в 24-часовом формате (01–24)	08	<a href="#">DT TO STRING FORMAT</a> , <a href="#">TOD TO STRING FORMAT</a>
hh	Час в 12-часовом формате (01–12)	08 (и для 8-00, и для 20-00)	<a href="#">DT TO STRING FORMAT</a> , <a href="#">TOD TO STRING FORMAT</a>
m	Минуты (0 – 59)	8	<a href="#">DT TO STRING FORMAT</a> , <a href="#">TOD TO STRING FORMAT</a>
mm	Минуты с ведущим нулем (00–59)	08	<a href="#">DT TO STRING FORMAT</a> , <a href="#">TOD TO STRING FORMAT</a>
s	Секунды (0 – 59)	8	<a href="#">DT TO STRING FORMAT</a> , <a href="#">TOD TO STRING FORMAT</a>
ss	Секунды с ведущим нулем (00–59)	08	<a href="#">DT TO STRING FORMAT</a> , <a href="#">TOD TO STRING FORMAT</a>
t	Идентификатор для 12-часового формата: А (часы < 12) и Р (часы > 12)	А (8 часов)	<a href="#">DT TO STRING FORMAT</a> , <a href="#">TOD TO STRING FORMAT</a>
tt	Идентификатор для 12-часового формата: АМ (часы < 12) и РМ (часы > 12)	РМ (15 часов)	<a href="#">DT TO STRING FORMAT</a> , <a href="#">TOD TO STRING FORMAT</a>