

СПК

Описание библиотеки OwenStringUtils

Версия: 1.3
Дата: 12.12.2017



Оглавление

1. Цель документа	3
2. Описание библиотеки OwenStringUtils.....	3
2.1. Установка библиотеки	3
2.2. Добавление библиотеки в проект CODESYS.....	4
2.3. Описание библиотеки.....	5
2.3.1. Функция CP1251_TO_UNICODE	5
2.3.2. Функция UNICODE_TO_CP1251	6
2.3.3. Функция Before	7
2.3.4. Функция WBefore.....	8
2.3.5. Функция After	9
2.3.6. Функция WAfter	10
2.3.7. Функция Between.....	11
2.3.8. Функция WBetween	12
2.3.9. Функция LowerCase.....	13
2.3.10. Функция WLowerCase	14
2.3.11. Функция UpperCase	15
2.3.12. Функция WUpperCase.....	16
2.3.13. Функция DT_STRING.....	17
2.3.14. Функция REAL_TO_STRING_FORMAT	18
2.3.15. Функция LREAL_TO_STRING_FORMAT	19

1. Цель документа

Данный документ 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 **3.5.4.6**.

Обратите внимание, что 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** выберите пункт **Репозиторий библиотек**, нажмите кнопку **Установить**, укажите путь к библиотеке и нажмите **Открыть**:

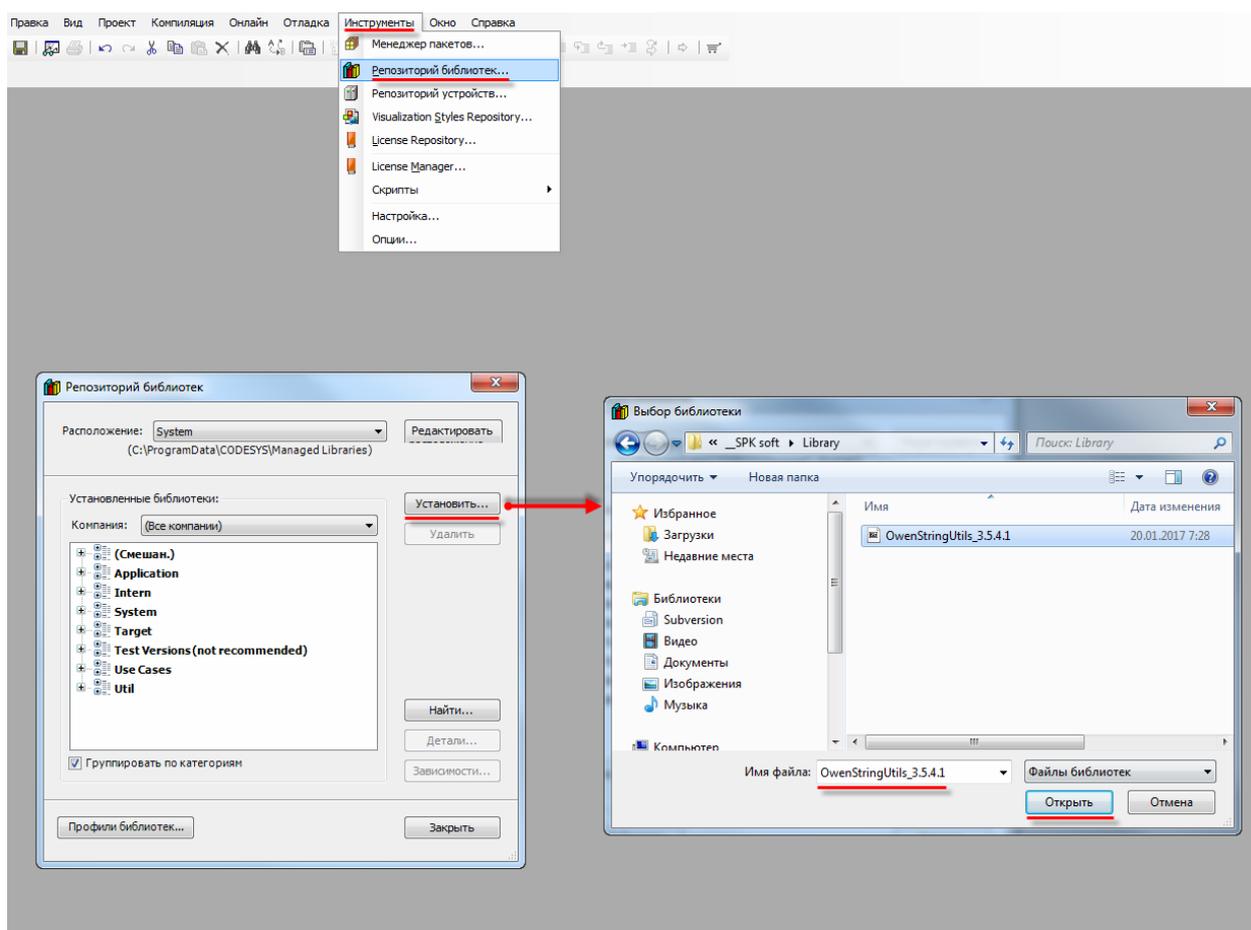


Рис. 2.1. Установка библиотеки в среду **CODESYS**

2.2. Добавление библиотеки в проект CODESYS

Для добавления библиотеки **OwenStringUtils** в проект **CODESYS**, в **Менеджере библиотек** нажмите кнопку **Добавить библиотеку** и в строке поиска введите **OwenStringUtils**, после чего выберите из списка нужную библиотеку и нажмите **OK**.

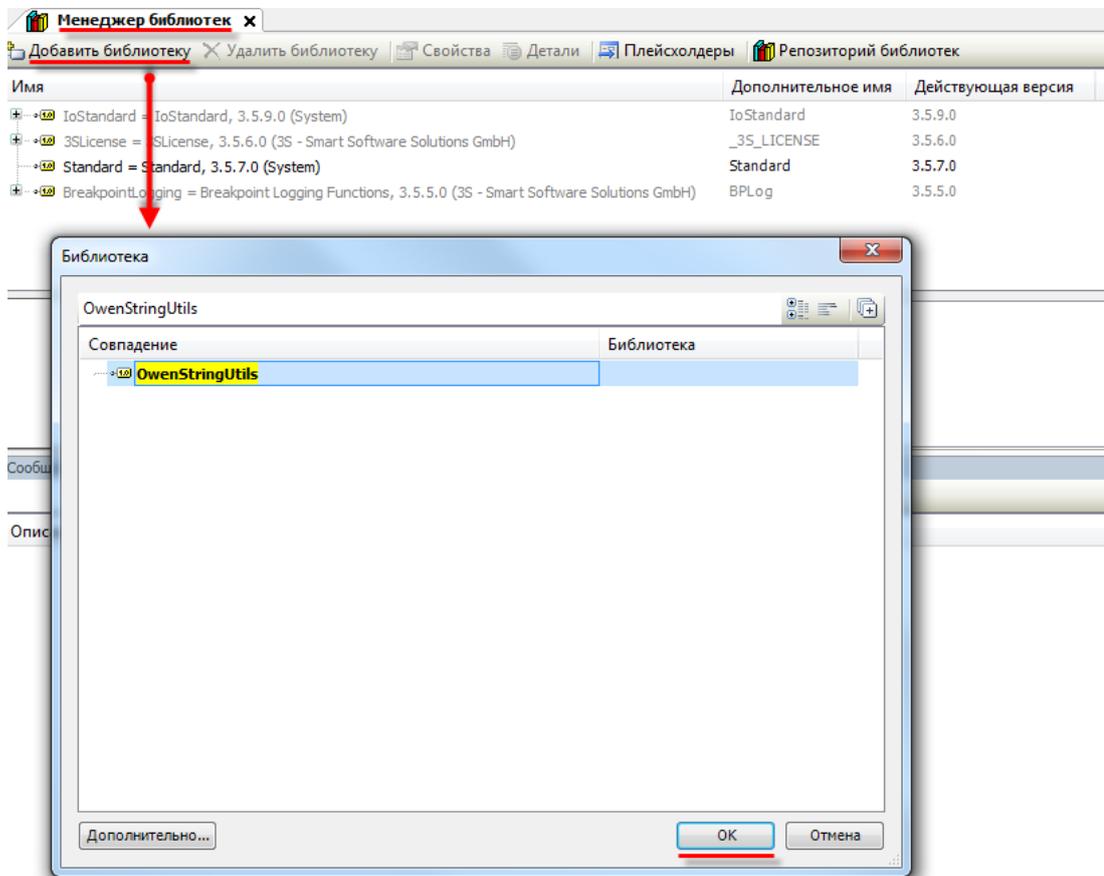


Рис. 2.2. Добавление библиотеки **OwenStringUtils**

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

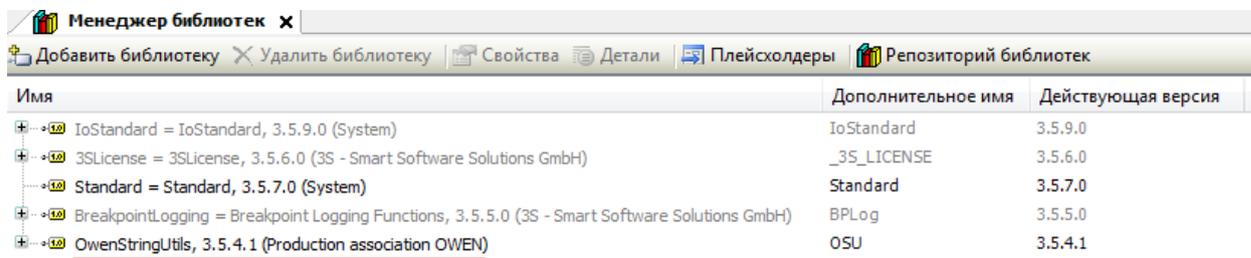


Рис. 2.3. Список библиотек проекта

Обратите внимание, что при обращении к ФБ и функциям библиотеки необходимо 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 \(UCS-2\)](#).

Имя переменной	Тип	Описание
Входные переменные		
sInputString	STRING	Исходная строка в кодировке ASCII (CP1251) .
Выходные переменные		
CP1251_TO_UNICODE	WSTRING	Строка в кодировке Unicode (UCS-2) .

Expression	Type	Value
sVar	STRING	'Съешь же еще этих мягких французских булок да выпей чаю'
wsVar	WSTRING	"Съешь же еще этих мягких французских булок да выпей чаю"

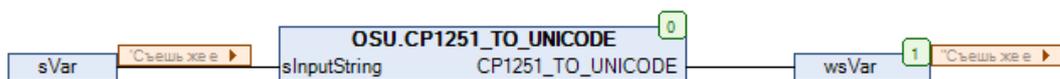


Рис. 2.4. Пример использования функции **CP1251_TO_UNICODE** на языке **CFC**

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 \(CP1251\)](#)

Имя переменной	Тип	Описание
Входные переменные		
wsInputString	WSTRING	Исходная строка в кодировке Unicode (UCS-2) .
Выходные переменные		
UNICODE_TO_CP1251	STRING	Строка в кодировке ASCII (CP1251) .

Expression	Type	Value
wsVar	WSTRING	"Съешь же еще этих мягких французских булок да выпей чаю"
sVar	STRING	'Съешь же еще этих мягких французских булок да выпей чаю'

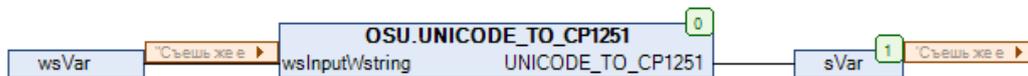


Рис. 2.5. Пример использования функции **UNICODE_TO_CP1251** на языке **CFC**

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**.

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

Expression	Type	Value
sSource	STRING	'Съешь же еще этих мягких французских булок да выпей чаю'
sPostfix	STRING	'же'
sBefore	STRING	'Съешь'



Рис. 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**.

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

Expression	Type	Value
wsSource	WSTRING	"Съешь же еще этих мягких французских булок да выпей чаю"
wsPostfix	WSTRING	" же "
wsBefore	WSTRING	"Съешь"



Рис. 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**.

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

Expression	Type	Value
sSource	STRING	'Съешь же еще этих мягких французских булок да выпей чаю'
sPrefix	STRING	'выпей '
sAfter	STRING	'чаю'

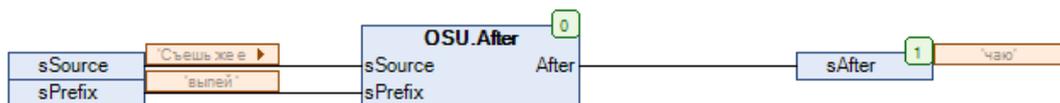


Рис. 2.8. Пример использования функции **After** на языке **CFC**

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**.

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

Expression	Type	Value
wsSource	WSTRING	"Съешь же еще этих мягких французских булок да выпей чаю"
wsPrefix	WSTRING	"выпей "
wsAfter	WSTRING	"чаю"



Рис. 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**.

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

Expression	Type	Value
◆ sSource	STRING	'Съешь же еще этих мягких французских булок да выпей чаю'
◆ sPrefix	STRING	'мягких '
◆ sPostfix	STRING	' да'
◆ sBetween	STRING	'французских булок'

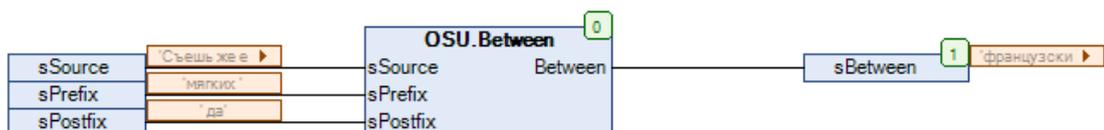


Рис. 2.10. Пример использования функции **Between** на языке **CFC**

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**.

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

Expression	Type	Value
wsSource	WSTRING	"Съешь же еще этих мягких французских булок да выпей чаю"
wsPrefix	WSTRING	"мягких "
wsPostfix	WSTRING	" да"
wsBetween	WSTRING	"французских булок"

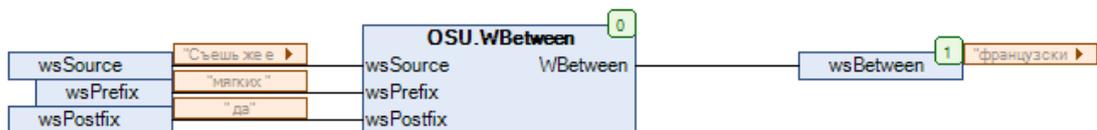


Рис. 2.11. Пример использования функции **WBetween** на языке **CFC**

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**.

Имя переменной	Тип	Описание
Входные переменные		
sStringToConvert	STRING	Исходная строка.
Выходные переменные		
LowerCase	STRING	Строка в нижнем регистре.

Выражение	Тип	Значение
sStringToConvert	STRING	'Съешь же еще этих МЯГКИХ ФРАНЦУЗКИХ БУЛОК да выпей чаю'
sLowerCase	STRING	'съешь же еще этих мягких французских булок да выпей чаю'

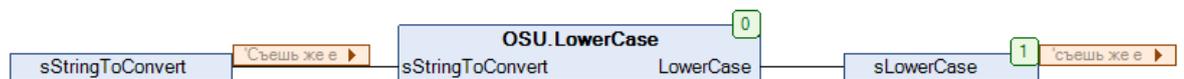


Рис. 2.12. Пример использования функции **LowerCase** на языке **CFC**

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**.

Имя переменной	Тип	Описание
Входные переменные		
wsStringToConvert	WSTRING	Исходная строка.
Выходные переменные		
WLowerCase	WSTRING	Строка в нижнем регистре.

Выражение	Тип	Значение
wsStringToConvert	WSTRING	"Съешь же еще этих МЯГКИХ ФРАНЦУЗКИХ БУЛОК да выпей чаю"
wsLowerCase	WSTRING	"съешь же еще этих мягких французских булок да выпей чаю"



Рис. 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**.

Имя переменной	Тип	Описание
Входные переменные		
sStringToConvert	STRING	Исходная строка.
Выходные переменные		
UpperCase	STRING	Строка в верхнем регистре.

Выражение	Тип	Значение
sStringToConvert	STRING	'Съешь же еще этих МЯГКИХ ФРАНЦУЗКИХ БУЛОК да выпей чаю'
sUpperCase	STRING	'СЪЕШЬ ЖЕ ЕЩЕ ЭТИХ МЯГКИХ ФРАНЦУЗКИХ БУЛОК ДА ВЫПЕЙ ЧАЮ'



Рис. 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**.

Имя переменной	Тип	Описание
Входные переменные		
wsStringToConvert	WSTRING	Исходная строка.
Выходные переменные		
WUpperCase	WSTRING	Строка в верхнем регистре.

Выражение	Тип	Значение
wsStringToConvert	WSTRING	"Съешь же еще этих МЯГКИХ ФРАНЦУЗКИХ БУЛОК да выпей чаю"
wsUpperCase	WSTRING	"СЪЕШЬ ЖЕ ЕЩЕ ЭТИХ МЯГКИХ ФРАНЦУЗКИХ БУЛОК ДА ВЫПЕЙ ЧАЮ"



Рис. 2.15. Пример использования функции **WUpperCase** на языке **CFC**

2.3.13. Функция DT_STRING

Функция **DT_STRING** преобразует метку времени типа **DT** в форматированную строку типа **STRING**.

Имя переменной	Тип	Описание
Входные переменные		
dtToConvert	DT	Метка времени.
Выходные переменные		
DT_STRING	STRING(19)	Метка времени в виде форматированной строки.

Выражение	Тип	Значение
dtDateTime	DATE_AND_TIME	DT#2017-12-5-11:2:33
sDateTime	STRING(19)	'2017-12-05 11:02:33'

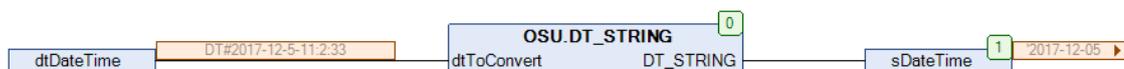


Рис. 2.16. Пример использования функции **DT_STRING** на языке **CFC**

2.3.14. 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**. При выборе недопустимого символа в качестве разделителя используется точка.

Имя переменной	Тип	Описание
Входные переменные		
rValueToConvert	REAL	Значение с плавающей точкой.
usiSignificantDigitsCount	USINT	Кол-во знаков после разделителя.
eDecimalSeparator	DECIMAL_SEPARATOR	Разделитель целой и дробной части.
Выходные переменные		
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'

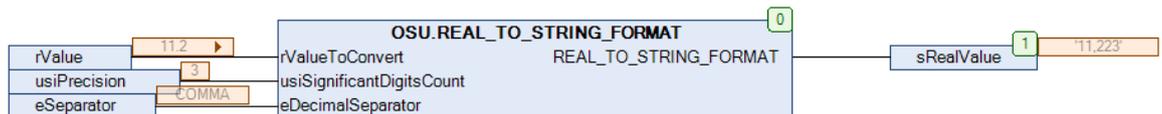


Рис. 2.17. Пример использования функции **REAL_TO_STRING_FORMAT** на языке CFC

2.3.15. 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**. При выборе недопустимого символа в качестве разделителя используется точка.

Имя переменной	Тип	Описание
Входные переменные		
lrValueToConvert	LREAL	Значение с плавающей точкой.
usiSignificantDigitsCount	USINT	Кол-во знаков после разделителя.
eDecimalSeparator	DECIMAL_SEPARATOR	Разделитель целой и дробной части.
Выходные переменные		
LREAL_TO_STRING_FORMAT	STRING(80)	Значение в виде форматированной строки.

Выражение	Тип	Значение
lValue	LREAL	11.223344556677
usiPrecision	USINT	8
eSeparator	DECIMAL_SEPARATOR	COMMA
sRealValue	STRING	'11,22334456'

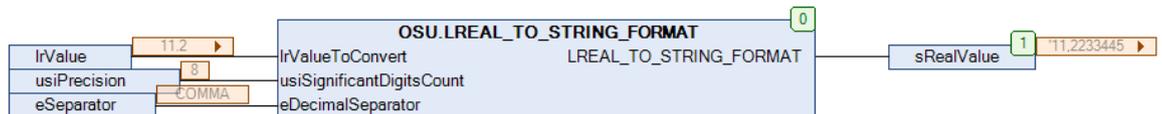


Рис. 2.18. Пример использования функции **LREAL_TO_STRING_FORMAT** на языке **CFC**