

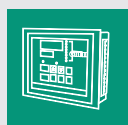
2017



СПК

Описание библиотеки CmpSysExec

Версия: 1.1
Дата: 15.03.2017



Оглавление

1. Цель документа	3
2. Описание библиотеки CmpSysExec	3
2.1. Установка библиотеки	3
2.2. Добавление библиотеки в проект CODESYS.....	5
2.3. Описание библиотеки	6
2.3.1. Функция SysExecute	6
2.3.2. ФБ SysExecute2	7

1. Цель документа

Данный документ представляет собой описание библиотеки **CmpSysExec**, которая позволяет организовать доступ к терминалу ОС **Linux** из программы контроллера. Реализация библиотеки находится в **Linux** (библиотека **CODESYS** представляет собой только интерфейс), поэтому может меняться в зависимости от версии прошивки контроллера. В данном документе описана версия библиотеки для прошивок СПК версии **4.800** и выше.

Обратите внимание, что библиотека может использоваться только контроллерами, программируемыми в среде **CODESYS 3.5** (например, СПК и ПЛК323).

2. Описание библиотеки CmpSysExec

2.1. Установка библиотеки

Библиотека **CmpSysExec** доступна на диске с ПО, входящем в комплект поставки, а также на сайте компании [OBEH](#) в разделе **CODESYS V3/Библиотеки**. Она распространяется как отдельно, так и в составе пакета библиотек Овен **LibInstall**.

Для установки пакета в **CODESYS** в меню **Инструменты** выберите пункт **Менеджер пакетов**, после чего укажите путь к файлу пакета и нажмите **Установить**:

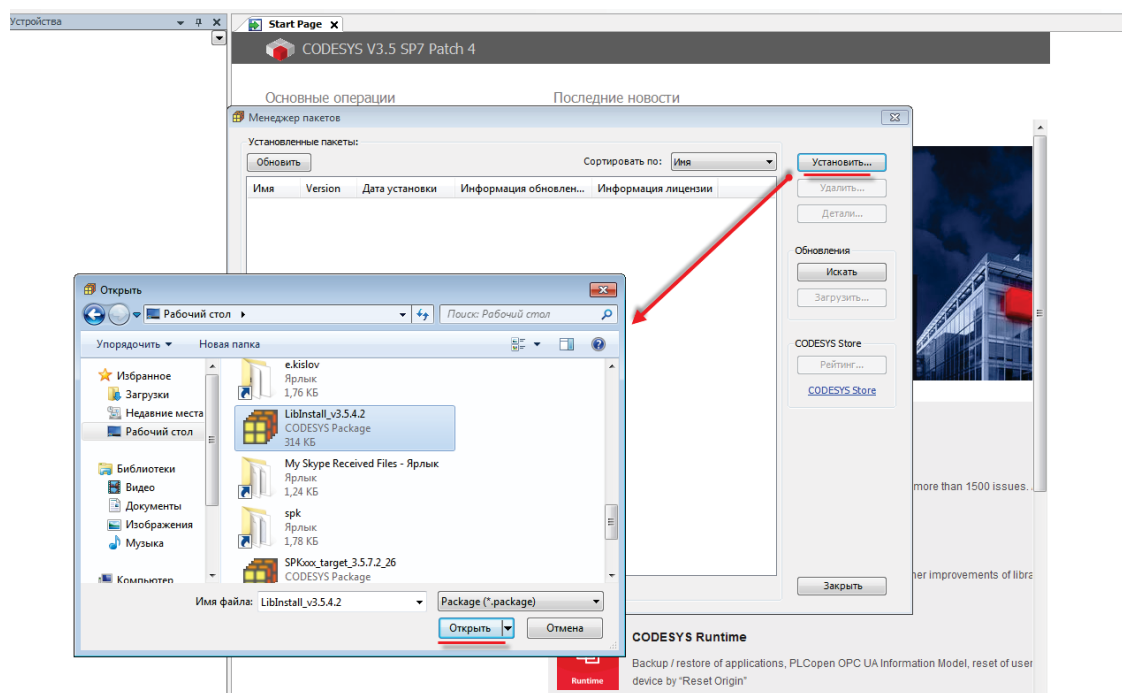


Рис. 2.1. Установка пакета библиотек Овен в среду **CODESYS**

В появившемся диалоговом окне выберите пункт **Полная установка**, после чего нажмите кнопку **Next**:

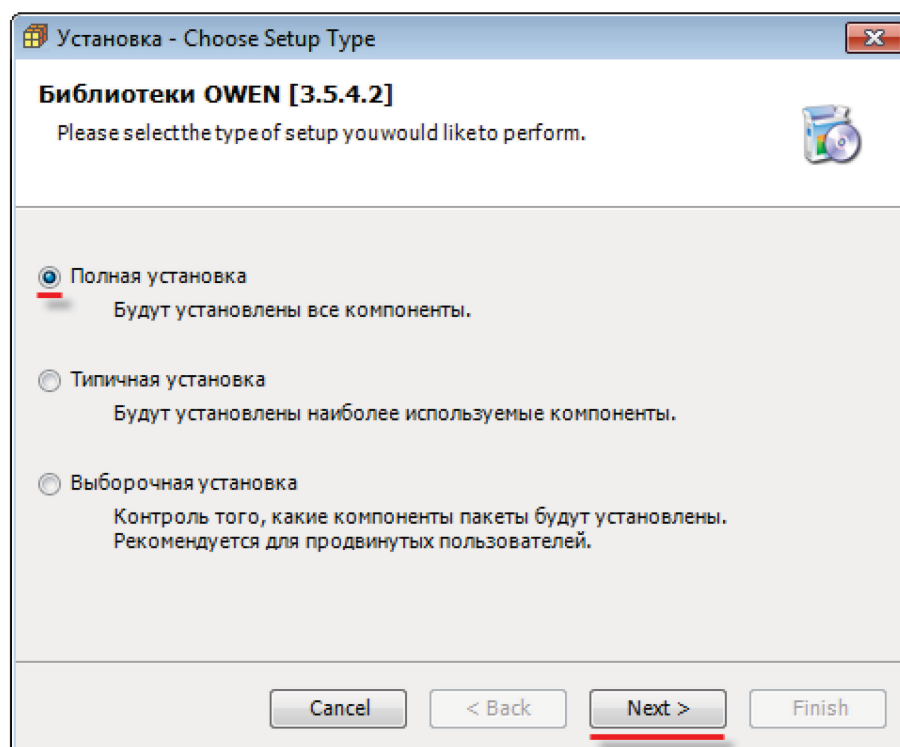


Рис. 2.2. Начало установки пакета библиотек

После завершения установки закройте диалоговое окно с помощью кнопки **Finish**:

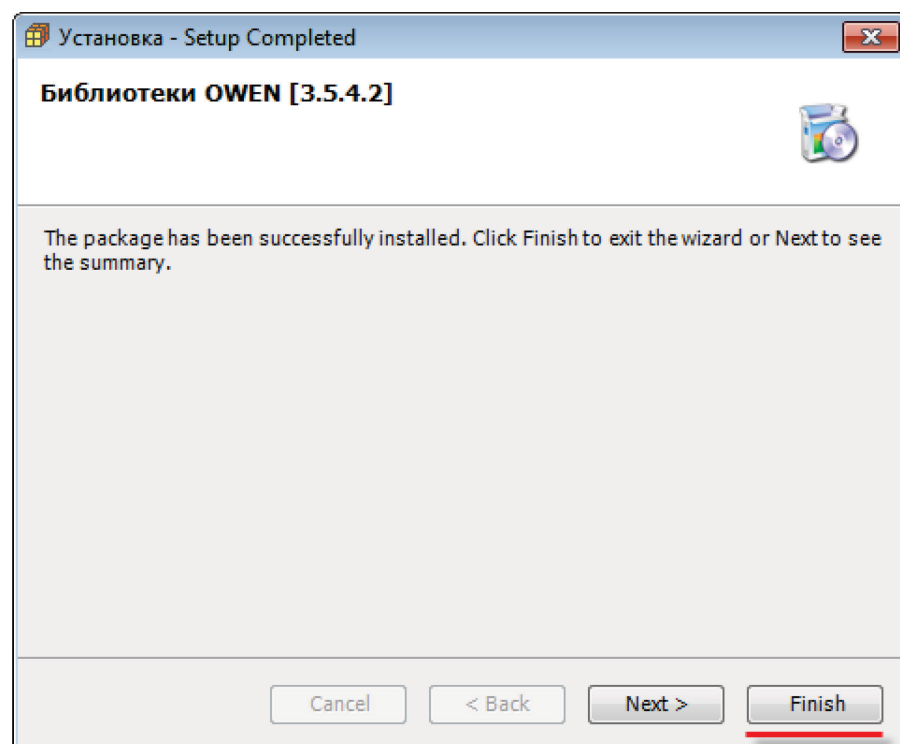


Рис. 2.3. Завершение установки пакета библиотек

2.2. Добавление библиотеки в проект CODESYS

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

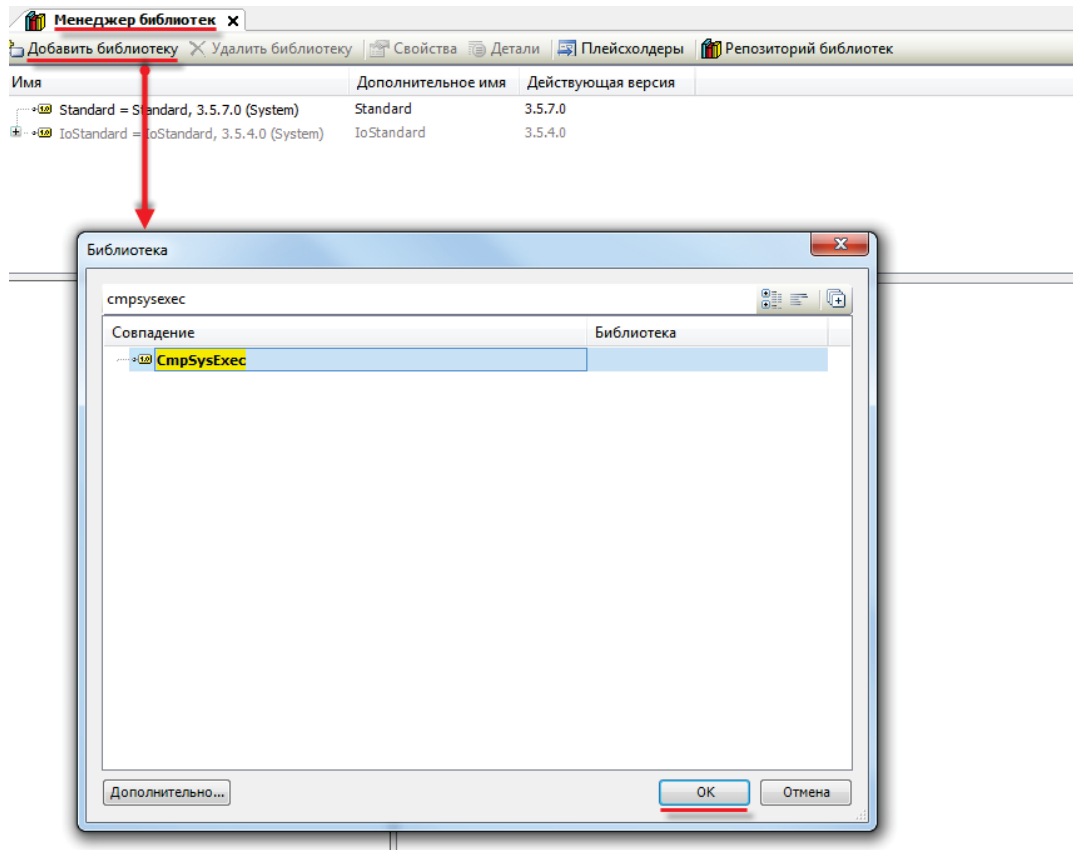


Рис. 2.4. Добавление библиотеки **CmpSysExec**

После добавления библиотеки появятся в списке **Менеджера библиотек**:

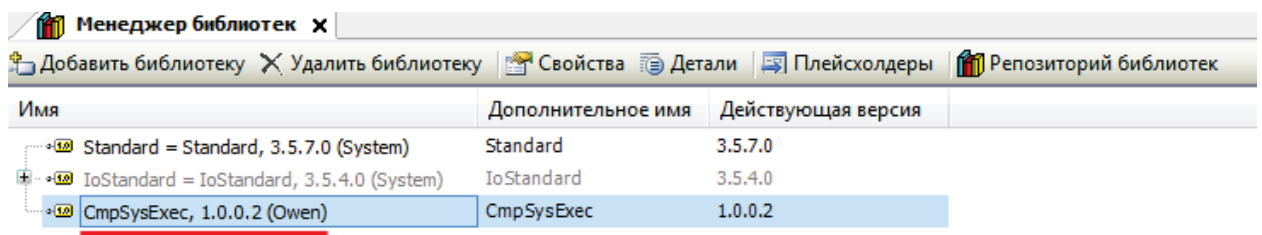


Рис. 2.5. Список библиотек проекта

2.3. Описание библиотеки

2.3.1. Функция SysExecute

Функция **SysExecute** используется для отправки команды в терминал **Linux**.

Функция выполняется в блокирующем режиме – иными словами, она занимает ПОТОК на все время выполнения, в связи с чем работа проекта **CODESYS** на это время останавливается.

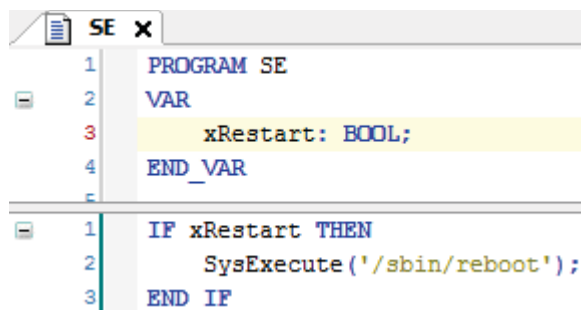
Для запуска функции в фоновом режиме следует после текста команды добавить символ **&** (например, 'echo "test" &').

Обратите внимание, что функция **SysExecute** должна вызываться исключительно по триггеру, а не в цикле программы.

Имя переменной	Тип	Описание
Входные переменные		
sys_command	STRING	Команда, подаваемая в терминал Linux.

На рис. 2.6 приведен пример использования функции. Когда переменная **xRestart** примет значение **TRUE**, в терминал будет отправлена команда **/sbin/reboot**, в результате чего контроллер перезагрузится.

Обратите внимание, что при отладке функции **SysExecute** настоятельно рекомендуется не создавать загрузочного приложения – при некорректном использовании функции это может привести к негативным последствиям. Например, если в приведенном ниже коде переменная **xRestart** будет являться энергонезависимой (**retain**), то контроллер будет циклически перезагружаться.



```
SE x
1 PROGRAM SE
2 VAR
3   xRestart: BOOL;
4 END_VAR
5
6 IF xRestart THEN
7   SysExecute('/sbin/reboot');
8 END_IF
```

Рис. 2.6. Пример использования функции **SysExecute**

2.3.2. ФБ SysExecute2

Функциональный блок **SysExecute2** используется для отправки команды в терминал **Linux** и получения ответа.

ФБ выполняется в неблокирующем режиме – иными словами, он освобождает поток после подачи команды и через какое-то время забирает ее ответ.

Команда ФБ выполняется с наименьшим приоритетом (**nice -n 20**).

Обратите внимание, что количество одновременно (в пределах циклах) используемых экземпляров ФБ ограничено **9-ю**. В случае использования более 9-ти экземпляров ФБ, работать будут только первые 9 (в порядке выполнения).

Имя переменной	Тип	Описание
Входные переменные		
cmd	STRING	Команда, подаваемая в терминал Linux .
Выходные переменные		
output_string	STRING	Текущая строка ответа. После отправки команды (это может занять несколько циклов) в этой переменной последовательно будут появляться строки ответа. Поскольку обработка одной строки может занять несколько циклов, то в эту переменную могут быть записаны пустые строки, поэтому необходимо производить в программе соответствующую проверку (см. рис. 2.7).
complete	BOOL	Флаг завершения работы блока. Принимает значение TRUE на следующий цикл после получения последней строки ответа.

Пример работы с ФБ приведен на рис. 2.7.

Когда переменная **xEnable** принимает значение **TRUE**, то переменная **i** принимает значение **0** (позиция строки массива для записи) и в терминал отправляется команда **ifconfig** (выводит информацию о сетевых интерфейсах контроллера). Полученный ответ построчно записывается в массив переменных типа **STRING** с названием **asOutput**. После завершения работы ФБ входу **xEnable** присваивается значение **FALSE** для предотвращения циклического вызова.

```
PLC_PRG x
1 PROGRAM PLC_PRG
2
3 VAR CONSTANT
4   c_iOutputSize :INT :=20;
5 END_VAR
6
7 VAR
8   fb_SE2 :SysExecute2;
9   fb_Rtrig :R_TRIG;
10
11   xEnable :BOOL;
12   i :INT;
13   asOutput :ARRAY [0..c_iOutputSize] OF STRING;
14
15 END_VAR
16
17 fb_Rtrig(CLK:=xEnable);
18
19 IF fb_Rtrig.Q THEN
20   i:=0;
21 END_IF
22
23 IF xEnable THEN
24
25   fb_SE2(cmd:='ifconfig', output_string=>asOutput[i]);
26
27   IF (NOT(fb_SE2.complete)) AND (i<c_iOutputSize) AND (fb_SE2.output_string<>'') THEN
28     i:=i+1;
29   END_IF
30
31   xEnable:=NOT(fb_SE2.complete);
32
33 END_IF
```

Рис. 2.7. Пример использования ФБ **SysExecute2**

Device.Application.PLC_PRG		
Выражение	Тип	Значение
◆ c_OutputSize	INT	20
⊕ ◆ fb_SE2	SysExecute2	
◆ xEnable	BOOL	FALSE
⊖ ◆ asOutput	ARRAY [0..c_Output...	
◆ asOutput[0]	STRING	'eth0 Link encap:Ethernet Hwaddr 6A:77:00:82:02:6C \$N'
◆ asOutput[1]	STRING	' inet addr:10.2.11.20 Bcast:10.2.255.255 Mask:255.255.0.0\$N'
◆ asOutput[2]	STRING	' UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1\$N'
◆ asOutput[3]	STRING	' RX packets:172025 errors:0 dropped:34602 overruns:0 frame:0\$N'
◆ asOutput[4]	STRING	' TX packets:30322 errors:0 dropped:0 overruns:0 carrier:0\$N'
◆ asOutput[5]	STRING	' collisions:0 txqueuelen:1000 \$N'
◆ asOutput[6]	STRING	' RX bytes:18597634 (17.7 MiB) TX bytes:11533524 (10.9 MiB)\$N'
◆ asOutput[7]	STRING	'\$N'
◆ asOutput[8]	STRING	'lo Link encap:Local Loopback \$N'
◆ asOutput[9]	STRING	' inet addr:127.0.0.1 Mask:255.0.0.0\$N'
◆ asOutput[10]	STRING	' UP LOOPBACK RUNNING MTU:16436 Metric:1\$N'
◆ asOutput[11]	STRING	' RX packets:678 errors:0 dropped:0 overruns:0 frame:0\$N'
◆ asOutput[12]	STRING	' TX packets:678 errors:0 dropped:0 overruns:0 carrier:0\$N'
◆ asOutput[13]	STRING	' collisions:0 txqueuelen:0 \$N'
◆ asOutput[14]	STRING	' RX bytes:41116 (40.1 KiB) TX bytes:41116 (40.1 KiB)\$N'
◆ asOutput[15]	STRING	'\$N'
◆ asOutput[16]	STRING	'
◆ asOutput[17]	STRING	'
◆ asOutput[18]	STRING	'
◆ asOutput[19]	STRING	'
◆ asOutput[20]	STRING	'
◆ i	INT	0

Рис. 2.8. Результат выполнения команды **ifconfig** через ФБ **SysExecute2**

```

/mnt/ufs/root #
/mnt/ufs/root # ifconfig
eth0      Link encap:Ethernet  Hwaddr 6A:77:00:82:02:6C
          inet addr:10.2.11.20  Bcast:10.2.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:754347 errors:0 dropped:107823 overruns:0 frame:0
          TX packets:161454 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:75334802 (71.8 MiB)  TX bytes:62867814 (59.9 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1315 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1315 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:74240 (72.5 KiB)  TX bytes:74240 (72.5 KiB)

```

Рис. 2.9. Результат выполнения команды **ifconfig** в терминале **Linux**