

# Модуль подсистемы “Специальные” <FLibSYS>

Модуль:	FLibSYS
Имя:	Библиотека функций системного API.
Тип:	Специальные
Источник:	spec_FLibSYS.so
Версия:	1.0.0
Автор:	Роман Савоченко
Описание:	Предоставляет библиотеку системного API среды пользовательского программирования.
Лицензия:	GPL

## Оглавление

Модуль подсистемы “Специальные” <FLibSYS>.....	1
Введение.....	2
1. Общесистемные функции.....	2
1.1. Вызов консольных команд и утилит операционной системы (sysCall).....	2
1.2. SQL запрос (dbReqSQL).....	2
1.3. Узел XML (xmlNode).....	3
1.4. Запрос интерфейса управления (xmlCntrReq).....	3
1.5. Архив значений (vArh).....	4
1.6. Буфер архива значений (vArhBuf).....	4
2. Функции для работы с астрономическим временем.....	5
2.1. Строка времени (tmFStr) <3047>.....	5
2.2. Полная дата (tmDate) <973>.....	5
2.3. Абсолютное время (tmTime) <220>.....	5
2.4. Конвертация времени из символьного представления во время в секундах, от эпохи 1.1.1970 (tmStrPTime) <2600>.....	6
2.5. Планирование времени в формате Cron (tmCron).....	6
3. Функции работы с сообщениями.....	7
3.1. Запрос сообщений (messGet).....	7
3.2. Генерация сообщения (messPut).....	7
4. Функции работы с строками.....	8
4.1. Получение размера строки (strSize) <114>.....	8
4.2. Получение части строки (strSubstr) <413>.....	8
4.3. Вставка одной строки в другую (strInsert) <1200>.....	8
4.4. Замена части строки другой (strReplace) <531>.....	8
4.5. Разбор строки по разделителю (strParse) <537>.....	9
4.6. Разбор пути (strParsePath) <300>.....	9
4.7. Путь в строку с разделителем (strPath2Sep).....	9
4.8. Кодирование строки в HTML (strEnc2HTML).....	10
4.9. Кодирование текста в бинарный вид (strEnc2Bin).....	10
4.10. Декодирование текста из бинарного вида (strDec4Bin).....	10
4.11. Преобразование вещественного в строку (real2str).....	10
4.12. Преобразование целого в строку (int2str).....	10
4.13. Преобразование строки в вещественное (str2real).....	11
4.14. Преобразование строки в целое (str2int).....	11
5. Функции работы с вещественным.....	12
5.1. Разделение float на слова (floatSplitWord) <56>.....	12
5.2. Объединение float из слов (floatMergeWord) <70>.....	12
6. API пользовательского программирования.....	13

# Введение

Специальный модуль FLibSYS предоставляет в систему OpenSCADA статическую библиотеку функций для работы с системой OpenSCADA, на уровне её системного API. Эти функции могут использоваться в среде пользовательского программирования системы OpenSCADA для организации неординарных алгоритмов взаимодействия.

Для адресации к функциям этой библиотеки можно использовать статический адрес вызова `"Special.FLibSYS.{Func}()` или динамический `"SYS.Special.FLibSYS["{Func}"].call()`, `"SYS.Special.FLibSYS.{Func}()`. Где *{Func}* — идентификатор функции в библиотеке.

Ниже приведено описание каждой функции библиотеки. Для каждой функции производилась оценка времени исполнения. Измерение производилось на системе со следующими параметрами: Athlon 64 3000+ (ALTLinux 4.0(32бит)) путём замера общего времени исполнения функции при вызове её 1000 раз. Выборка производилась из пяти вычислений с округлением до меньшего целого. Время заключается в угловые скобки и измеряется в микросекундах.

## 1. Общесистемные функции

### 1.1. Вызов консольных команд и утилит операционной системы (sysCall)

*Описание:* Осуществляет вызовы консольных команд ОС. Функция открывает широкие возможности пользователю OpenSCADA путём вызова любых системных программ, утилит и скриптов, а также получения посредством них доступа к огромному объёму системных данных. Например команда “ls -l” вернёт детализированное содержимое рабочей директории.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
com	Команда	Строка	Вход	

*Пример:*

```
using Special.FLibSYS;  
test=sysCall("ls -l");  
messPut("Example",0,"Example: "+test);
```

### 1.2. SQL запрос (dbReqSQL)

*Описание:* Формирование SQL-запроса к БД.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Объект(Массив)	Возврат	
addr	Адрес БД	Строка	Вход	
req	SQL-запрос	Строка	Вход	

### 1.3. Узел XML (xmlNode)

*Описание:* Создание объекта узла XML.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Объект(XMLNodeObj)	Возврат	
name	Имя	Строка	Вход	

*Пример:*

```
using Special.FLibSYS;
//Создание объекта "get" узла XML.
req = xmlNode("get");
//Создание объекта "get" узла XML с созданием атрибутов.
//sub_DAQ/mod_ModBus/cntr_1/prm_1 - путь согласно структуре проекта
req = xmlNode("get").setAttr("path", "/sub_DAQ/mod_ModBus/cntr_1/prm_1/%2fprm
%2fst%2fen");
```

### 1.4. Запрос интерфейса управления (xmlCntrReq)

*Описание:* Запрос интерфейса управления к системе посредством XML. Обычный запрос записывается в виде `<get path="/OPath/%2felem"/>`. При указании станции осуществляется запрос к внешней станции.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
req	Запрос	Объект(XMLNodeObj)	Выход	
stat	Станция	Строка	Вход	

*Пример:*

```
using Special.FLibSYS;
//Получение признака "Включен/Выключен" параметра "1" контроллера "1" модуля
"ModBus".
//sub_DAQ/mod_ModBus/cntr_1/prm_1 - путь согласно структуре проекта
req = xmlNode("get").setAttr("path", "/sub_DAQ/mod_ModBus/cntr_1/prm_1/%2fprm
%2fst%2fen");
rez = xmlCntrReq(req);
messPut("test", 0, "Значение: "+req.text());

//Установка признака "Включен" параметра "1" контроллера "1" модуля "ModBus".
req = xmlNode("set").setAttr("path", "/sub_DAQ/mod_ModBus/cntr_1/prm_1/%2fprm
%2fst%2fen").setText(1);
rez = xmlCntrReq(req);

//Установка признака "Выключен" параметра "1" контроллера "1" модуля "ModBus".
req = xmlNode("set").setAttr("path", "/sub_DAQ/mod_ModBus/cntr_1/prm_1/%2fprm
%2fst%2fen").setText(0);
rez = xmlCntrReq(req);
```

## 1.5. Архив значений (vArh)

*Описание:* Получение объекта архива значений (VArchObj) путём подключения к архиву по его адресу.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Объект(VArchObj)	Возврат	
name	Имя, адрес к атрибуту параметра с архивом (DAQ.{Module}.{Cntr}.{Prm}.{Attr}) или непосредственно к архиву значений (Archive.{ValArchive}).	Строка	Вход	

### Объект VArchObj

*Функции:*

- *begin( usec, archivor )* — Получение времени начала архива путём возврата секунд и микросекунд <usec> для архиватора <archivor>.
- *end( usec, archivor )* — Получение времени окончания архива путём возврата секунд и микросекунд <usec> для архиватора <archivor>.
- *period( usec, archivor )* — Получение периодичности архива путём возврата секунд и микросекунд <usec> для архиватора <archivor>.
- *get( sec, usec, upOrd, archivor )* — Получение значения из архива на время <sec>:<usec> с привязкой к верху <upOrd> и для архиватора <archivor>. Реальное время полученного значения устанавливается в <sec>:<usec>.
- *set( val, sec, usec )* — Запись значения <val> в буфер архива на время <sec>:<usec>.
- *copy( src, begSec, begUSec, endSec, endUSec, archivor )* — Копирование части исходного <src> архива или его буфера в текущий начиная с <begSec>:<begUSec> и заканчивая <endSec>:<endUSec> для архиватора <archivor>.
- *FFT( tm, size, archivor, tm\_usec )* — Выполняет разложение в ряд Фурье с помощью FFT алгоритма. Возвращается массив амплитуд частот для окна значений из архива с временем начала <tm>:<tm\_usec> (секунды:микросекунды), глубиной в историю архива <size> (секунд) и для архиватора <archivor>.

*Пример:*

```
using Special.FLibSYS;
val = vArh(strPath2Sep(addr)).get(time,uTime,0,archtor);
return val.isEval() ? "Пусто" : real2str(val,prec);
```

## 1.6. Буфер архива значений (vArhBuf)

*Описание:* Получение объекта буфера архива значений (VArchObj) для выполнения промежуточных операций над кадрами данных.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Объект(VArchObj)	Возврат	
tp	Тип значений архива (0-Boolean, 1-Integer, 4-Real, 5-String)	Целый	Вход	1
sz	Максимальный размер буфера	Целый	Вход	100
per	Периодичность буфера (в микросекундах)	Целый	Вход	1000000
hgrd	Режим «Жесткая сетка времени»	Логический	Вход	0
hres	Режим «Высокого разрешения времени (микросекунды)»	Логический	Вход	0

## 2. Функции для работы с астрономическим временем

### 2.1. Строка времени (tmFStr) <3047>

*Описание:* Преобразует абсолютное время в строку нужного формата. Запись формата соответствует POSIX-функции strftime.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
val	Строка полной даты	Строка	Возврат	
sec	Секунды	Целый	Вход	0
form	Формат	Строка	Вход	%Y-%m-%d %H:%M:%S

*Пример:*

```
using Special.FLibSYS;  
test=tmFStr(SYS.time(),"%d %m %Y");  
messPut("Example",0,"tmFStr(): "+test);
```

### 2.2. Полная дата (tmDate) <973>

*Описание:* Возвращает полную дату в секундах, минутах, часах и т.д, исходя из абсолютного времени в секундах от эпохи 1.1.1970.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
fullsec	Полные секунды	Целый	Вход	0
sec	Секунды	Целый	Выход	0
min	Минуты	Целый	Выход	0
hour	Часы	Целый	Выход	0
mday	День месяца	Целый	Выход	0
month	Месяц	Целый	Выход	0
year	Год	Целый	Выход	0
wday	День недели	Целый	Выход	0
yday	День в году	Целый	Выход	0
isdst	Daylight saving time	Целый	Выход	0

*Пример:*

```
using Special.FLibSYS;  
curMin=curHour=curDay=curMonth=curYear=0;  
tmDate(tmTime(),0,curMin,curHour,curDay,curMonth,curYear);  
messPut("test",0,"Текущая минута: "+curMin);  
messPut("test",0,"Текущий час : "+curHour);  
messPut("test",0,"Текущий день: "+curDay);  
messPut("test",0,"Текущий месяц: "+curMonth);  
messPut("test",0,"Текущий год: "+curYear);
```

### 2.3. Абсолютное время (tmTime) <220>

*Описание:* Возвращает абсолютное время в секундах от эпохи 1.1.1970 и микросекундах, если <usec> установлен в неотрицательное значение.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
sec	Секунды	Целый	Возврат	0
usec	Микросекунды	Целый	Выход	-1

## 2.4. Конвертация времени из символьного представления во время в секундах, от эпохи 1.1.1970 (tmStrPTime) <2600>

*Описание:* Возвращает время в секундах от эпохи 1.1.1970, исходя из строковой записи времени, в соответствии с указанным шаблоном. Например, шаблону "%Y-%m-%d %H:%M:%S" соответствует время «2006-08-08 11:21:55». Описание формата шаблона можно получить из документации на POSIX-функцию "strptime".

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
sec	Секунды	Целый	Возврат	0
str	Строка даты	Строка	Вход	
form	Формат записи даты	Строка	Вход	%Y-%m-%d %H:%M:%S

*Пример:*

```
using Special.FLibSYS;
curMin=curHour=curDay=curMonth=curYear=0;
tmDate(tmTime(),0,curMin,curHour,curDay,curMonth,curYear);
test = tmStrPTime(""+curYear+"-"+(curMonth+1)+"-"+curDay+" 9:0:0", "%Y-%m-%d %H:
%M:%S");
messPut("Example",0,"tmStrPTime(): "+test);
```

## 2.5. Планирование времени в формате Cron (tmCron)

*Описание:* Возвращает время спланированное в формате стандарта Cron начиная от базового времени или от текущего если базовое не указано.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
res	Результат	Целый	Возврат	0
str	Запись в стандарте Cron	Строка	Вход	* * * * *
base	Базовое время	Целый	Вход	0

## 3. Функции работы с сообщениями

### 3.1. Запрос сообщений (messGet)

*Описание:* Запрос системных сообщений.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Объект(Массив)	Возврат	
btm	Время начала	Целое	Вход	
etm	Время конца	Целое	Вход	
cat	Категория сообщения	Строка	Вход	
lev	Уровень сообщения	Целый	Вход	
arch	Архиватор	Строка	Вход	

### 3.2. Генерация сообщения (messPut)

*Описание:* Формирование системного сообщения.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
cat	Категория сообщения	Строка	Вход	
lev	Уровень сообщения	Целый	Вход	
mess	Текст сообщения	Строка	Вход	

*Пример:*

```
rnd_sq_gr11_lineClr="red";  
Special.FLibSYS.messPut("Example",1,"Event: "+rnd_sq_gr12_leniClr);
```

## 4. Функции работы с строками

### 4.1. Получение размера строки (strSize) <114>

*Описание:* Используется для получения размера.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Целый	Возврат	
str	Строка	Строка	Вход	

*Пример:*

```
Special.FLibSYS.messPut("Example",1,"ReturnString: "+strSize("Example"));
```

### 4.2. Получение части строки (strSubstr) <413>

*Описание:* Используется для получения части строки.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
str	Строка	Строка	Вход	
pos	Позиция	Целый	Вход	0
n	Количество	Целый	Вход	-1

*Пример:*

```
using Special.FLibSYS;  
test=strSubstr("Example", 0, strSize("Example")-1);  
messPut("Example",1,"ReturnString: "+test);
```

### 4.3. Вставка одной строки в другую (strInsert) <1200>

*Описание:* Используется для вставки одной строки в другую.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
str	Строка	Строка	Выход	
pos	Позиция	Целый	Вход	0
ins	Вставляемая строка	Строка	Вход	

### 4.4. Замена части строки другой (strReplace) <531>

*Описание:* Используется для замены части строки другой строкой.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
str	Строка	Строка	Выход	
pos	Позиция	Целый	Вход	0
n	Количество	Целый	Вход	-1
rep1	Заменяющая строка	Строка	Вход	

## 4.5. Разбор строки по разделителю (strParse) <537>

*Описание:* Используется в разборе строки по разделителю.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
str	Строка	Строка	Вход	
lev	Уровень	Целый	Вход	
sep	Разделитель	Строка	Вход	","
off	Смещение	Целый	Выход	

*Пример:*

```
using Special.FLibSYS;
ExampleString="Example:123";
test=strParse(ExampleString,1,":");
messPut("Example",0,"strParse(): "+test);
```

## 4.6. Разбор пути (strParsePath) <300>

*Описание:* Используется в разборе пути на элементы.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
path	Путь	Строка	Вход	
lev	Уровень	Целый	Вход	
off	Смещение	Целый	Выход	

*Пример:*

```
using Special.FLibSYS;
test=strParsePath(path,0,"/");
messPut("Example",1,"strParsePath(): "+test);
```

## 4.7. Путь в строку с разделителем (strPath2Sep)

*Описание:* Используется для преобразования пути в строку с разделителем.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
src	Источник	Строка	Вход	
sep	Разделитель	Строка	Вход	","

*Пример:*

```
//Преобразуем значение "/ses_AGLKS/pg_so" атрибута "path" в значение
"ses_AGLKS.pg_so"
using Special.FLibSYS;
test = strPath2Sep(path);
messPut("Example",0,"path: "+path);
messPut("Example",0,"strPath2Sep(): "+test);
```

#### 4.8. Кодирование строки в HTML (strEnc2HTML)

*Описание:* Используется для кодирования строки для использования в исходнике HTML.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
src	Источник	Строка	Вход	

#### 4.9. Кодирование текста в бинарный вид (strEnc2Bin)

*Описание:* Используется для кодирования текста в бинарный вид, из формата <00 A0 FA DE>.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
src	Источник	Строка	Вход	

#### 4.10. Декодирование текста из бинарного вида (strDec4Bin)

*Описание:* Используется для декодирования текста из бинарного вида в формат <00 A0 FA DE>.

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
src	Источник	Строка	Вход	

#### 4.11. Преобразование вещественного в строку (real2str)

*Описание:* Используется для преобразования вещественного в строку.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
val	Значение	Вещественное	Вход	
prc	Точность	Целое	Вход	4
tp	Тип	Строка	Вход	"f"

#### 4.12. Преобразование целого в строку (int2str)

*Описание:* Используется для преобразования целого в строку.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Строка	Возврат	
val	Значение	Целое	Вход	
base	База, поддерживаются: 8, 10, 16	Целое	Вход	10

#### 4.13. Преобразование строки в вещественное (str2real)

*Описание:* Используется для преобразования строки в вещественное.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Вещественное	Возврат	
val	Значение	Строка	Вход	

#### 4.14. Преобразование строки в целое (str2int)

*Описание:* Используется для преобразования строки в целое.

*Параметры:*

ID	Имя	Тип	Режим	По умолчанию
rez	Результат	Целое	Возврат	
val	Значение	Строка	Вход	
base	Основа	Целый	Вход	0

## 5. Функции работы с вещественным

### 5.1. Разделение float на слова (floatSplitWord) <56>

*Описание:* Разделение float (4 байтов) на слова (2 байта).

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
val	Значение	Вещественное	Вход	
w1	Слово 1	Целый	Выход	
w2	Слово 2	Целый	Выход	

### 5.2. Объединение float из слов (floatMergeWord) <70>

*Описание:* Объединение float (4 байтов) из слов (2 байта).

*Параметры:*

ID	Параметр	Тип	Режим	По умолчанию
rez	Результат	Вещественное	Возврат	
w1	Слово 1	Целый	Вход	
w2	Слово 2	Целый	Вход	

## 6. API пользовательского программирования

Некоторые объекты модуля предоставляют функции пользовательского программирования.

### Объект "Библиотека функций" (SYS.Special.FLibSYS)

- *ELTp {funcID}(ELTp prm1, ...)* — вызов функции библиотеки *{funcID}*. Возвращает результат вызываемой функции.

### Объект "Пользовательская функция" (SYS.Special.FLibSYS["funcID"])

- *ELTp call(ELTp prm1, ...)* — вызов данной функции с параметрами  $\langle prm\{N\} \rangle$ . Возвращает результат вызываемой функции.