

Библиотека Standard AgavaSCADA/AgavaPLC 1.2+



Содержание

Общие положения

Standard - стандартная библиотека

Математические и тригонометрические функции

Преобразование IEEE битов

Сравнение с допуском

Тригонометрические функции

Гиперболические функции

Экспоненциальные и логарифмические функции

Степенные функции

Функции округления и остатка

Специализированные функции

Функции для работы с узлами

Функции для работы с окнами

Функции для работы с временем

Функции для работы с событиями

Функции для работы со строками

Другие функции

1 Общие положения

Данное описание предназначено для AgavaSCADA/AgavaPLC 1.2+

2 Standard - стандартная библиотека

2.1 Математические и тригонометрические функции

2.1.1 Преобразование IEEE битов

```
float fpFromIEEE(uint bits)
```

- **Назначение:** Преобразует 32-битное IEEE-754 представление в число float
- **Возвращаемое значение:** число float

- **Аргументы:** uint bits - битовое представление

```
| uint fpToIEEE(float value)
```

- **Назначение:** Преобразует число float в 32-битное IEEE-754 представление
- **Возвращаемое значение:** битовое представление (uint)
- **Аргументы:** float value - исходное число

```
| double fpFromIEEE(uint64 bits)
```

- **Назначение:** Преобразует 64-битное IEEE-754 представление в число double
- **Возвращаемое значение:** число double
- **Аргументы:** uint64 bits - битовое представление

```
| uint64 fpToIEEE(double value)
```

- **Назначение:** Преобразует число double в 64-битное IEEE-754 представление
- **Возвращаемое значение:** битовое представление (uint64)
- **Аргументы:** double value - исходное число

2.1.2 Сравнение с допуском

```
| bool closeTo(float a, float b, float epsilon = 0.00001f)
```

- **Назначение:** Сравнивает два числа float с заданной погрешностью
- **Возвращаемое значение:** true, если числа близки в пределах погрешности; иначе false
- **Аргументы:**
 - float a - первое число;
 - float b - второе число;
 - float epsilon (опционально) - погрешность (по умолчанию 0.00001)

```
| bool closeTo(double a, double b, double epsilon = 0.0000000001)
```

- **Назначение:** Сравнивает два числа double с заданной погрешностью
- **Возвращаемое значение:** true, если числа близки в пределах погрешности; иначе false
- **Аргументы:**
 - double a - первое число;
 - double b - второе число;
 - double epsilon (опционально) - погрешность (по умолчанию 1e-10)

2.1.3 Тригонометрические функции

```
| float cos(float angle)
```

- **Назначение:** Вычисляет косинус угла
- **Возвращаемое значение:** косинус угла
- **Аргументы:** float angle - угол в радианах

```
| float sin(float angle)
```

- **Назначение:** Вычисляет синус угла
- **Возвращаемое значение:** синус угла
- **Аргументы:** float angle - угол в радианах

```
| float tan(float angle)
```

- **Назначение:** Вычисляет тангенс угла
- **Возвращаемое значение:** тангенс угла
- **Аргументы:** float angle - угол в радианах

```
| float acos(float value)
```

- **Назначение:** Вычисляет арккосинус (обратный косинус)
- **Возвращаемое значение:** угол в радианах
- **Аргументы:** float value - значение в диапазоне [-1, 1]

```
| float asin(float value)
```

- **Назначение:** Вычисляет арксинус (обратный синус)
- **Возвращаемое значение:** угол в радианах
- **Аргументы:** float value - значение в диапазоне [-1, 1]

```
| float atan(float value)
```

- **Назначение:** Вычисляет арктангенс (обратный тангенс)
- **Возвращаемое значение:** угол в радианах
- **Аргументы:** float value - значение

```
| float atan2(float y, float x)
```

- **Назначение:** Вычисляет арктангенс двух переменных
- **Возвращаемое значение:** угол в радианах
- **Аргументы:**
 - float y - координата Y;

- float x - координата X

2.1.4 Гиперболические функции

```
| float cosh(float x)
```

- **Назначение:** Вычисляет гиперболический косинус
- **Возвращаемое значение:** гиперболический косинус
- **Аргументы:** float x - значение

```
| float sinh(float x)
```

- **Назначение:** Вычисляет гиперболический синус
- **Возвращаемое значение:** гиперболический синус
- **Аргументы:** float x - значение

```
| float tanh(float x)
```

- **Назначение:** Вычисляет гиперболический тангенс
- **Возвращаемое значение:** гиперболический тангенс
- **Аргументы:** float x - значение

2.1.5 Экспоненциальные и логарифмические функции

```
| float log(float x)
```

- **Назначение:** Вычисляет натуральный логарифм (по основанию e)
- **Возвращаемое значение:** натуральный логарифм
- **Аргументы:** float x - значение > 0

```
| float log10(float x)
```

- **Назначение:** Вычисляет десятичный логарифм (по основанию 10)
- **Возвращаемое значение:** десятичный логарифм
- **Аргументы:** float x - значение > 0

2.1.6 Степенные функции

```
| float pow(float base, float exponent)
```

- **Назначение:** Возводит число в степень
- **Возвращаемое значение:** результат возведения в степень
- **Аргументы:**

- float base - основание;
- float exponent - показатель степени

```
| float sqrt(float x)
```

- **Назначение:** Вычисляет квадратный корень
- **Возвращаемое значение:** квадратный корень
- **Аргументы:** float x - неотрицательное значение

2.1.7 Функции округления и остатка

```
| float ceil(float x)
```

- **Назначение:** Округляет вверх до ближайшего целого
- **Возвращаемое значение:** округлённое вверх значение
- **Аргументы:** float x - значение

```
| float abs(float x)
```

- **Назначение:** Вычисляет абсолютное значение (модуль)
- **Возвращаемое значение:** абсолютное значение
- **Аргументы:** float x - значение

```
| float floor(float x)
```

- **Назначение:** Округляет вниз до ближайшего целого
- **Возвращаемое значение:** округлённое вниз значение
- **Аргументы:** float x - значение

```
| float fraction(float x)
```

- **Назначение:** Выделяет дробную часть числа
- **Возвращаемое значение:** дробная часть числа
- **Аргументы:** float x - значение

2.2 Специализированные функции

Специализированные функции среды разработки, доступные для использования в скриптах, приведены ниже.

2.2.1 Функции для работы с узлами

```
| float GetNodeValueAsFloat(string strNodePath)
```

- **Назначение:** Получить значение узла в виде float по заданному пути
- **Возвращаемое значение:** значение узла (float)
- **Аргументы:** string strNodePath - путь к узлу

```
void SetNodeValueAsFloat(string strNodePath, float fValue)
```

- **Назначение:** Установить значение узла в float по заданному пути
- **Возвращаемое значение:** нет
- **Аргументы:**
 - string strNodePath - путь к узлу;
 - float fValue - устанавливаемое значение

```
int GetNodeValueAsInt(string strNodePath)
```

- **Назначение:** Получить значение узла в виде int по заданному пути
- **Возвращаемое значение:** значение узла (int)
- **Аргументы:** string strNodePath - путь к узлу

```
void SetNodeValueAsInt(string strNodePath, int iValue)
```

- **Назначение:** Установить значение узла в int по заданному пути
- **Возвращаемое значение:** нет
- **Аргументы:**
 - string strNodePath - путь к узлу;
 - int iValue - устанавливаемое значение

```
string GetNodeValueAsString(string strNodePath)
```

- **Назначение:** Получить значение узла в виде string по заданному пути
- **Возвращаемое значение:** значение узла (string)
- **Аргументы:** string strNodePath - путь к узлу

```
void SetNodeValueAsString(string strNodePath, string strValue)
```

- **Назначение:** Установить значение узла в виде string по заданному пути
- **Возвращаемое значение:** нет
- **Аргументы:**
 - string strNodePath - путь к узлу;
 - string strValue - устанавливаемое значение

```
bool GetNodeValueAsBool(string strNodePath)
```

- **Назначение:** Получить значение узла в виде bool по заданному пути
- **Возвращаемое значение:** значение узла (bool)
- **Аргументы:** string strNodePath - путь к узлу

```
| void SetNodeValueAsBool(string strNodePath, bool bValue)
```

- **Назначение:** Установить значение узла в виде bool по заданному пути
- **Возвращаемое значение:** нет
- **Аргументы:**
 - string strNodePath - путь к узлу;
 - bool bValue - устанавливаемое значение

```
| bool NodeValueIsError(string strNodePath)
```

- **Назначение:** Проверка значения узла на ошибку
- **Возвращаемое значение:** true, если значение узла содержит ошибку; иначе false
- **Аргументы:** string strNodePath - путь к узлу

```
| void StartNode(string strNodePath)
```

- **Назначение:** Запуск узла по заданному пути
- **Возвращаемое значение:** нет
- **Аргументы:** string strNodePath - путь к узлу

```
| void StopNode(string strNodePath)
```

- **Назначение:** Остановка узла по заданному пути
- **Возвращаемое значение:** нет
- **Аргументы:** string strNodePath - путь к узлу

2.2.2 Функции для работы с окнами

```
| void CloseWindow(string strNodePath, int iDisplay)
```

- **Назначение:** Закрыть окно по заданному пути на дисплее с номером iDisplay
- **Возвращаемое значение:** нет
- **Аргументы:**
 - string strNodePath - путь к окну;
 - int iDisplay - номер дисплея

```
| void ShowWindow(string strNodePath, int iDisplay)
```

- **Назначение:** Отобразить окно по заданному пути на дисплее с номером iDisplay.

Координаты — абсолютные, установленные в свойствах окна.

- **Возвращаемое значение:** нет
- **Аргументы:**
 - string strNodePath - путь к окну;
 - int iDisplay - номер дисплея

```
| int LoadComposition(string strWindowPath, string strCompositionPath, int iDisplay)
```

- **Назначение:** Загрузить композицию strCompositionPath в окно strWindowPath на дисплее с номером iDisplay
- **Возвращаемое значение:** код результата загрузки
- **Аргументы:**
 - string strWindowPath - путь к окну;
 - string strCompositionPath - путь к композиции;
 - int iDisplay - номер дисплея
- **Примечания:** функция доступна с версии AS 1.6.25+

2.2.3 Функции для работы с временем

```
| int GetCurrentTime()
```

- **Назначение:** Получение текущего времени в UNIX формате (количество секунд с 01 января 1970 года)
- **Возвращаемое значение:** время в секундах (int)
- **Аргументы:** нет

```
| uint64 GetCurrentTimeMs()
```

- **Назначение:** Получение текущего времени в UNIX формате (количество миллисекунд с 01 января 1970 года)
- **Возвращаемое значение:** время в миллисекундах (uint64)
- **Аргументы:** нет
- **Примечания:** функция доступна с версии AS 1.2.55+, AS 1.6.1+

```
| int GetLocalTime()
```

- **Назначение:** Получение текущего времени в UNIX формате (количество секунд с 01 января 1970 года) с учётом часового пояса
- **Возвращаемое значение:** локальное время в секундах (int)
- **Аргументы:** нет

```
| int GetHours(int iTime)
```

- **Назначение:** Получение текущего часа из времени в формате UNIX
- **Возвращаемое значение:** час (int)
- **Аргументы:** int iTime - время в формате UNIX

```
int GetMinutes(int iTime)
```

- **Назначение:** Получение текущих минут из времени в формате UNIX
- **Возвращаемое значение:** минуты (int)
- **Аргументы:** int iTime - время в формате UNIX

```
int GetSeconds(int iTime)
```

- **Назначение:** Получение текущих секунд из времени в формате UNIX
- **Возвращаемое значение:** секунды (int)
- **Аргументы:** int iTime - время в формате UNIX

2.2.4 Функции для работы с событиями

```
int StoreMessage(int iMessageLevel, string strMessage)
```

- **Назначение:** Генерация информационного события (типа «сообщение»). Событие помещается в группу «Системные».
- **Возвращаемое значение:** 0 — успешно, иначе код ошибки
- **Аргументы:**
 - int iMessageLevel - уровень сообщения (1 — наивысший): 1=FATAL, 2=ERROR, 3=WARNING, 4=NOTICE, 5=INFO, 6=TRACE, 7=DEBUG;
 - string strMessage - текст сообщения
- **Примечания:** функция доступна с версии AS 1.2.34+

```
int StoreMessage(int iMessageLevel, string strMessage, string strGroup)
```

- **Назначение:** Генерация информационного события (типа «сообщение»). Событие помещается в указанную при вызове группу.
- **Возвращаемое значение:** 0 — успешно, иначе код ошибки
- **Аргументы:**
 - int iMessageLevel - уровень сообщения (1 — наивысший): 1=FATAL, 2=ERROR, 3=WARNING, 4=NOTICE, 5=INFO, 6=TRACE, 7=DEBUG;
 - string strMessage - текст сообщения;
 - string strGroup - имя группы, в которую должно помещаться событие
- **Примечания:** функция доступна с версии AS 1.6.20+

2.2.5 Функции для работы со строками

```
| string EncodeMD5(string strText)
```

- **Назначение:** Кодирование строки в MD5 хэш
- **Возвращаемое значение:** MD5 хэш параметра
- **Аргументы:** string strText - исходная строка
- **Примечания:** функция доступна с версии AS 1.2.34+

2.2.6 Другие функции

```
| bool isnan(double)
```

- **Назначение:** Проверка на nan
- **Возвращаемое значение:** true, если значение является nan; иначе false
- **Аргументы:** double - проверяемое значение

```
| int LaunchProcess(string process)
```

- **Назначение:** Запуск процесса в операционной системе. Запуск производится в блокирующем режиме, то есть возврат из функции осуществляется по завершению процесса.
- **Возвращаемое значение:** код завершения процесса
- **Аргументы:** string process - команда/путь запускаемого процесса

Источник —

https://docs.kb-agava.ru/index.php?title=Библиотека_Standard_AgavaSCADA/AgavaPLC_1.2%2B&oldid=3677

Эта страница в последний раз была отредактирована 11 августа 2026 в 14:28.