Государственная корпорация по атомной энергии «Росатом»

**АО «Атомтехэнерго»**

Московский филиал «Центратомтехэнерго» АО «Атомтехэнерго»

**ШЛЮЗ ИНФОРМАЦИОННОГО ОБМЕНА**

**МЕЖДУ ЭМУЛЯТОРОМ ТПТС (ЭМТ) И SCADA АТЭ**

**Программное обеспечение**

**Руководство системного программиста**

**РУ АТЭ.0800.2921-2022**

**Листов 28**

2022

АННОТАЦИЯ

Настоящее руководство системного программиста предназначено для персонала, обеспечивающего поддержку и сопровождение программного обеспечения (ПО) шлюза информационного обмена между эмулятором ТПТС (ЭМТ) и SCADA АТЭ (далее – шлюз), входящего в состав расчетно-аналитического (экспериментального) комплекса пусконаладочных работ (РАЭК ПНР).

Настоящее руководство системного программиста включает в себя сведения для установки и настройки шлюза на узле ЭМТ, а также настройке программных средств на автоматизированном рабочем месте (АРМ) разработчика ЭМТ.

В данном документе приведены общие сведения о ПО с указанием назначения и функций, сведения о технических и системных программных средствах, обеспечивающих его выполнение, сведения о структуре, описание действий по настройке.

Системный программист, обеспечивающий поддержку программного обеспечения шлюза, должен обладать навыками работы в операционных системах (ОС) Microsoft Windows 10 и Astra Linux Special Edition 1.6, интегрированной среде разработки Microsoft Visual Studio 2017 (Visual C++) на уровне системного администратора.

СОДЕРЖАНИЕ

[1 Общие сведения 4](#_Toc112683561)

[2 Структура программы 5](#_Toc112683562)

[3 Настройка и подготовка программы к запуску 5](#_Toc112683563)

[4 Запуск и выполнение программы 31](#_Toc112683564)

[5 Сообщения системному программисту 31](#_Toc112683565)

[Перечень сокращений 41](#_Toc112683566)

[Лист изменений (дополнений) 42](#_Toc112683567)

# Общие сведения

## Назначение программного обеспечения

Шлюз предназначен для реализации задач сопряжения ЭМТ и SCADA АТЭ.

## Функции программного обеспечения

Основными функциями шлюза являются:

* обмен данными с ЭМТ по технологии разделяемой памяти;
* обмен данных со SCADA АТЭ по протоколу TCP/IP;
* мониторинг работы шлюза;
* предоставление информации о работе шлюза в виде записей в файл журнала регистрации на локальный диск ОС узла.

## Сведения о технических средствах

Шлюз предусматривает следующие требования к характеристикам технических средств узла:

* не менее 1 процессора Intel Core i5 5го поколения с частотой 2.4 ГГц и выше;
* оперативная память – 4 ГБ или более;
* свободное место на жестком диске – не менее 1 ГБ;
* наличие одного сетевого интерфейса 100/1000 Мбит/с;
* монитор разрешением 800x600 точек на дюйм или выше;
* манипулятор «мышь» и клавиатура.

Шлюз должен быть установлен на узел ЭМТ. Узлы ЭМТ и SCADA АТЭ должны находиться в одной сети.

## Сведения о программных средствах

Программное обеспечение АРМ разработчика ЭМТ должно предусматривать следующий минимальный состав программных средств, обеспечивающих выполнение программного обеспечения:

* операционная система Microsoft Windows 10 64 bit;
* среда разработки Microsoft Visual Studio 2017 (Visual C++).

Программное обеспечение узла ЭМТ должно предусматривать следующий минимальный состав программных средств, обеспечивающих выполнение программного обеспечения:

* операционная система Microsoft Windows 10 64 bit;
* программное обеспечение ЭМТ.

Программное обеспечение узла SCADA АТЭ должно предусматривать следующий минимальный состав программных средств, обеспечивающих выполнение программного обеспечения:

* операционная система Astra Linux Special Edition 1.6;
* ПО SCADA АТЭ.

# Структура программы

В состав программного обеспечения шлюза входят следующие элементы:

* программное обеспечение шлюза;
* библиотека для работы со шлюзом.

Схема информационного взаимодействия между ЭМТ, шлюзом и SCADA АТЭ представлена на рисунке Рисунок 1.

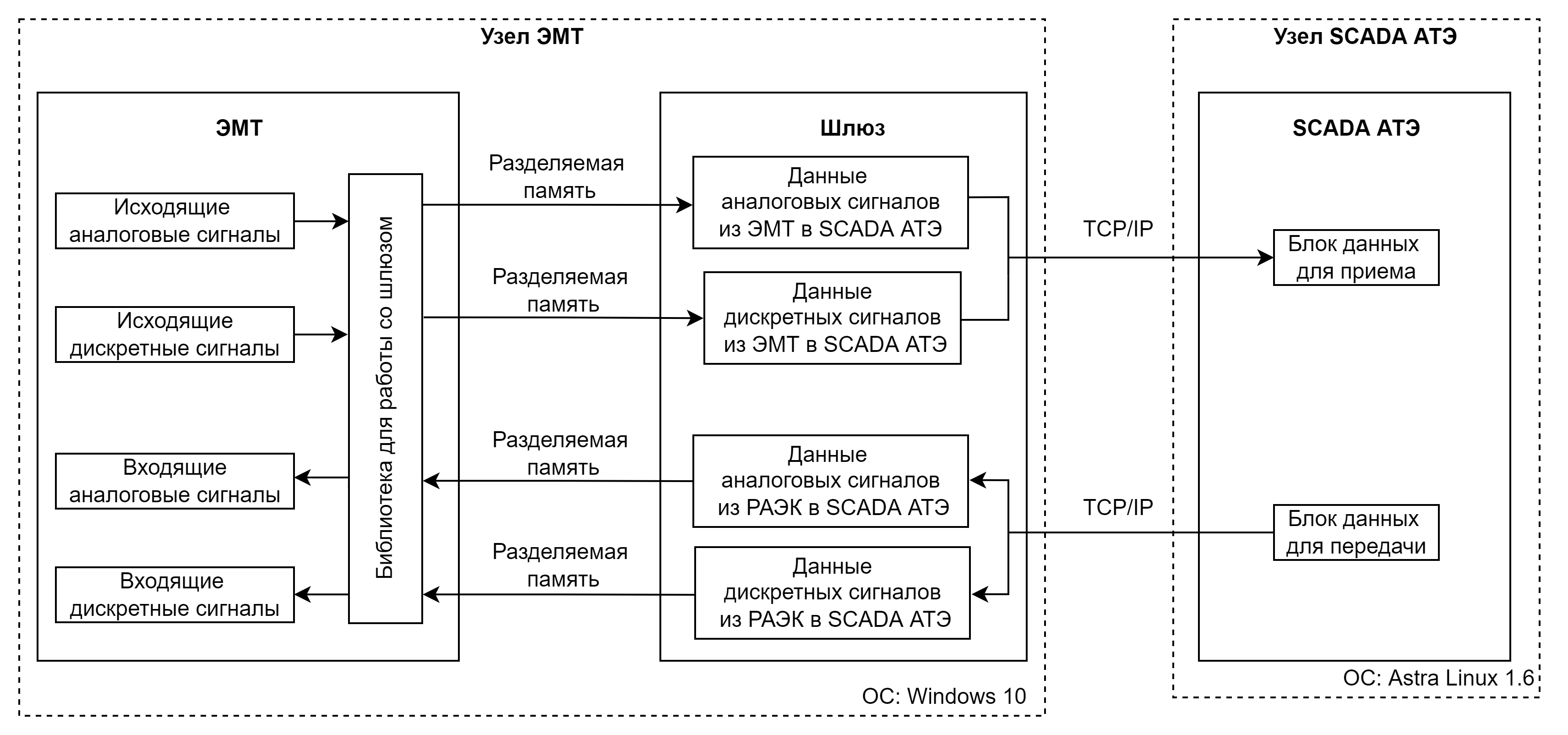


Рисунок 1 – Схема информационного взаимодействия

Обмен данными между программными средствами ЭМТ и шлюзом осуществляется по технологии разделяемой памяти посредством библиотеки «ClientSharedMemory.h», которая устанавливается на АРМ ЭМТ вместе с шлюзом.

Обмен данными между шлюзом и SCADA АТЭ осуществляется по протоколу TCP/IP. Шлюз считывает данные сигналов, записываемые ЭМТ, из разделяемой памяти. После чего шлюз отправляет данные на узел SCADA АТЭ. Частота передачи определяется настройками SCADA АТЭ.

# Настройка и подготовка программы к запуску

Для подготовки работы шлюза следует:

* установить библиотеку шлюза в ПО ЭМТ на АРМ разработчика ЭМТ;
* настроить проект ПО ЭМТ;
* установить полученное программное обеспечение на узел ЭМТ;
* установить шлюз на узел ЭМТ и выполнить его настройку.

## Последовательность установки и настройки

Установка программных средств на узел ЭМТ происходит в следующей последовательности:

* установка шлюза;
* настройка конфигурации работы шлюза;

Установка программных средств на АРМ разработчика ЭМТ происходит в следующей последовательности:

* установка библиотеки шлюза;
* настройка проекта ЭМТ.

Все этапы установки и настройки должны выполняться в указанной последовательности. Нарушение последовательности может повлечь за собой невозможность установки какого-либо из программных продуктов или итоговую неработоспособность системы в целом или какого-либо из ее компонентов.

## Установка и настройка программных средств узла ЭМТ

### Установка шлюза

Для установки шлюза запустите установочный файл с носителя с дистрибутивом (см. рисунок Рисунок 2).

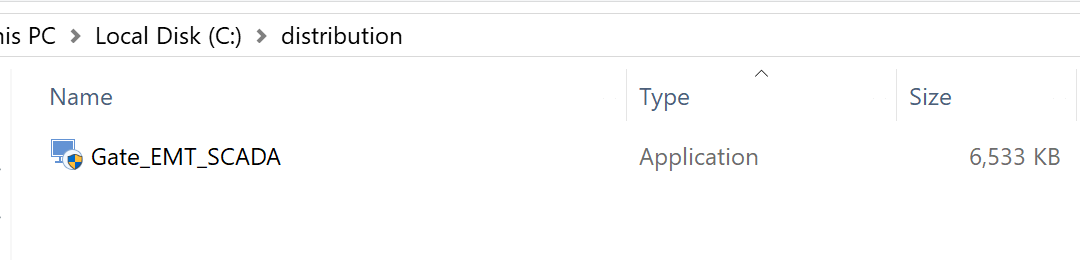


Рисунок – Запуск установки

В появившемся диалоговом окне мастера установки нажмите кнопку «Далее» (см. рисунок 3).

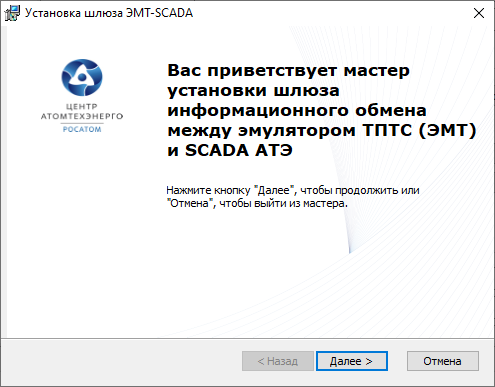


Рисунок 3 – Мастер установки

В появившемся диалоговом окне необходимо указать каталог установки либо оставить значение поля по умолчанию и нажать кнопку «Далее» (см. рисунок 4).

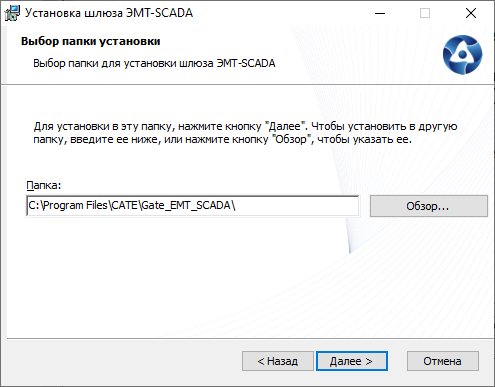


Рисунок 4 – Выбор каталога установки

В появившемся диалоговом окне необходимо подтвердить установку нажав кнопку «Установить» (см. рисунок 5).

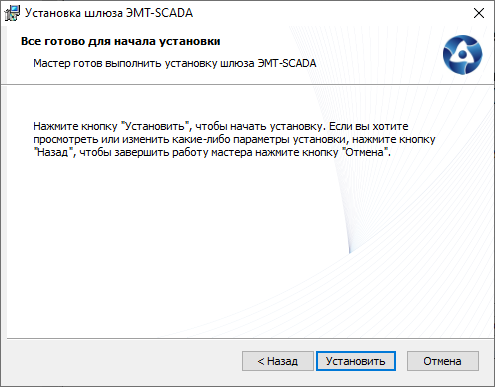


Рисунок 5 – Установка

Для окончания установки в появившемся диалоговом окне необходимо нажать кнопку «Готово» (см. рисунок 6).

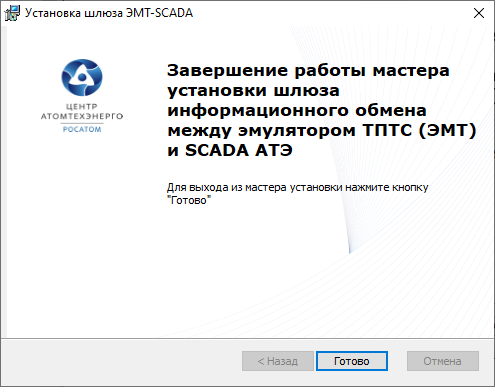


Рисунок 6 – Завершение установки

После завершения установки шлюз появится в списке зарегистрированных сервисов ОС (см. рисунок 7). На этом установка шлюза завершена.

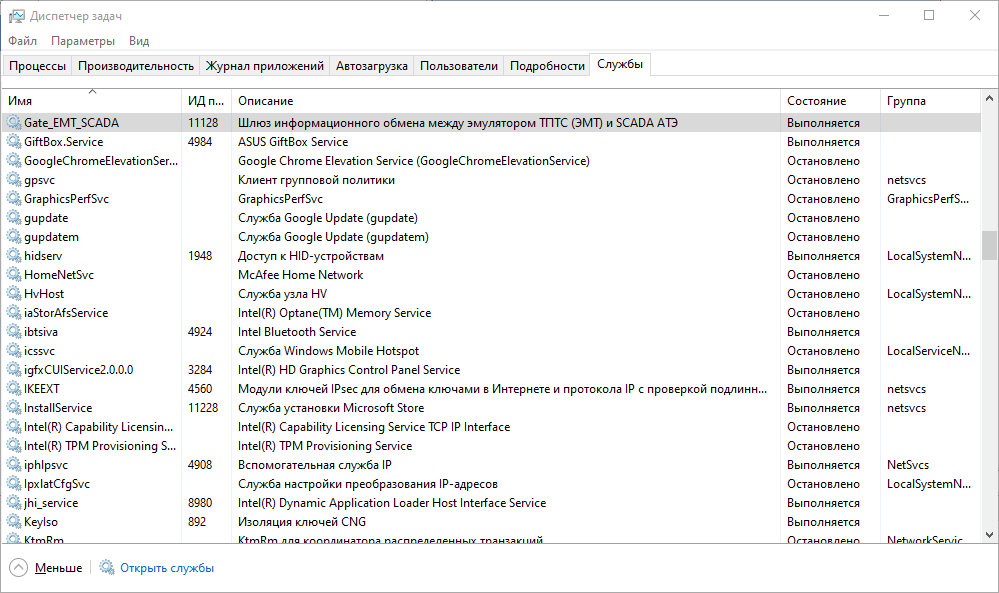


Рисунок 7 – Список зарегистрированных сервисов ОС

### Настройка конфигурации работы шлюза

Настройка параметров работы шлюза производится посредством ПО SCADA ATЭ. Файл конфигурации шлюза «config.json» хранит в себе последнюю установленную конфигурацию и располагаются в каталоге установки (по умолчанию C:\Program Files\CATE\Gate\_EMT\_SCADA) и включает в себя следующие разделы:

* dds – раздел конфигурации dds уровня;
* logger – раздел конфигурации логирования;
* adapters – раздел конфигурации адаптеров передачи данных (включающий в себя настройку разделяемой памяти и список тегов);
* другие служебные разделы.

Настройка списка передаваемых и принимаемых данных между SCADA АТЭ и ЭМТ также осуществляется посредством ПО SCADA АТЭ.

## Установка и настройка программных средств АРМ разработчика ЭМТ

### Установка библиотеки шлюза

Установка библиотеки шлюза производится совместно с установкой шлюза. Файл библиотеки шлюза расположен в каталоге установки (по умолчанию C:\Program Files\CATE\Gate\_EMT\_SCADA\EMT\) на локальном диске. Для разработки проекта необходимо включить в сборку файл «ClientSharedMemory.h» библиотеки в проект.

### Настройка проекта ЭМТ

Для настройки проекта ЭМТ необходимо перенести файл библиотеки в корневой каталок проекта ПО ЭМТ. Открыть проект ЭМТ в Visual Studio и подключить существующий заголовочный файл (см. рисунок 8).

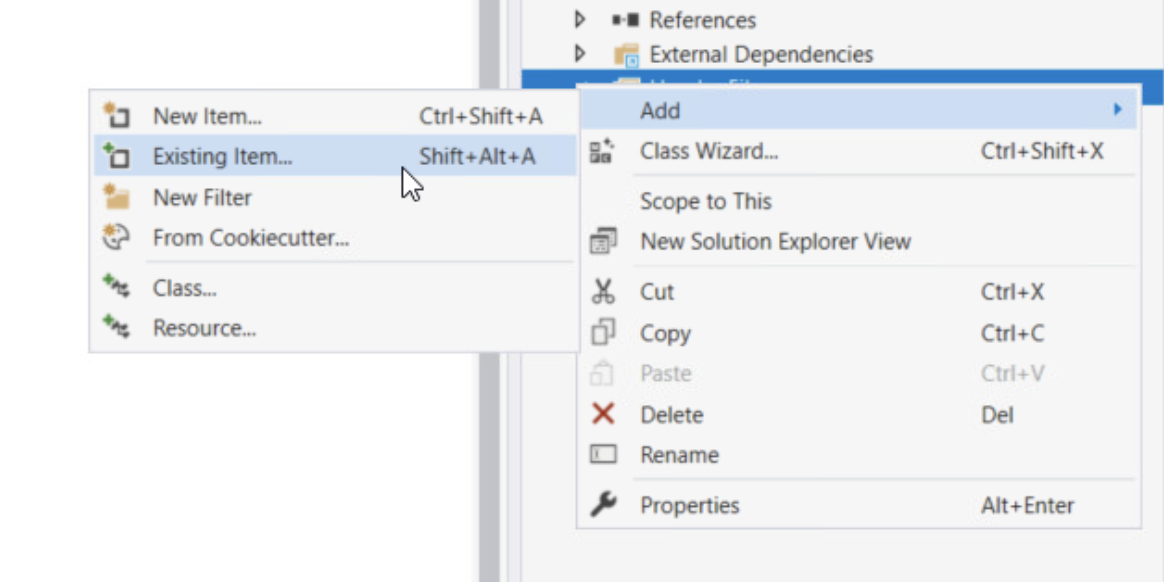


Рисунок 8 – Подключение заголовочного файла

Затем необходимо выбрать в корневом каталоге проекта файл «ClientSharedMemory.h» и нажать «Add». Или же можно добавить путь к библиотеке в список включаемых каталогов проекта.

Библиотека включает в себя пространство имен «scada\_ate::lib::emt», которое включает в себя:

* класс «ClientSharedMemory»: осуществляет взаимодействие с шлюзом;
* перечисление «ResultRequest»: определяет результат выполнения функций класса «ClientSharedMemory»;
* перечисление Status: определяет состояние объекта типа «ClientSharedMemory»;
* шаблонную структуру «Value»: определяет тип обмена данными с SCADA ATЭ. Шаблон включает в себя свойство value – определяет значение переменной, и quality – определяет качество переменной.

Класс «ClientSharedMemory» осуществляет функцию обмена данными с шлюзом посредством межпроцессорных коммуникаций, используя технологию разделяемой памяти (диаграмма класса представлена на рисунке 9).

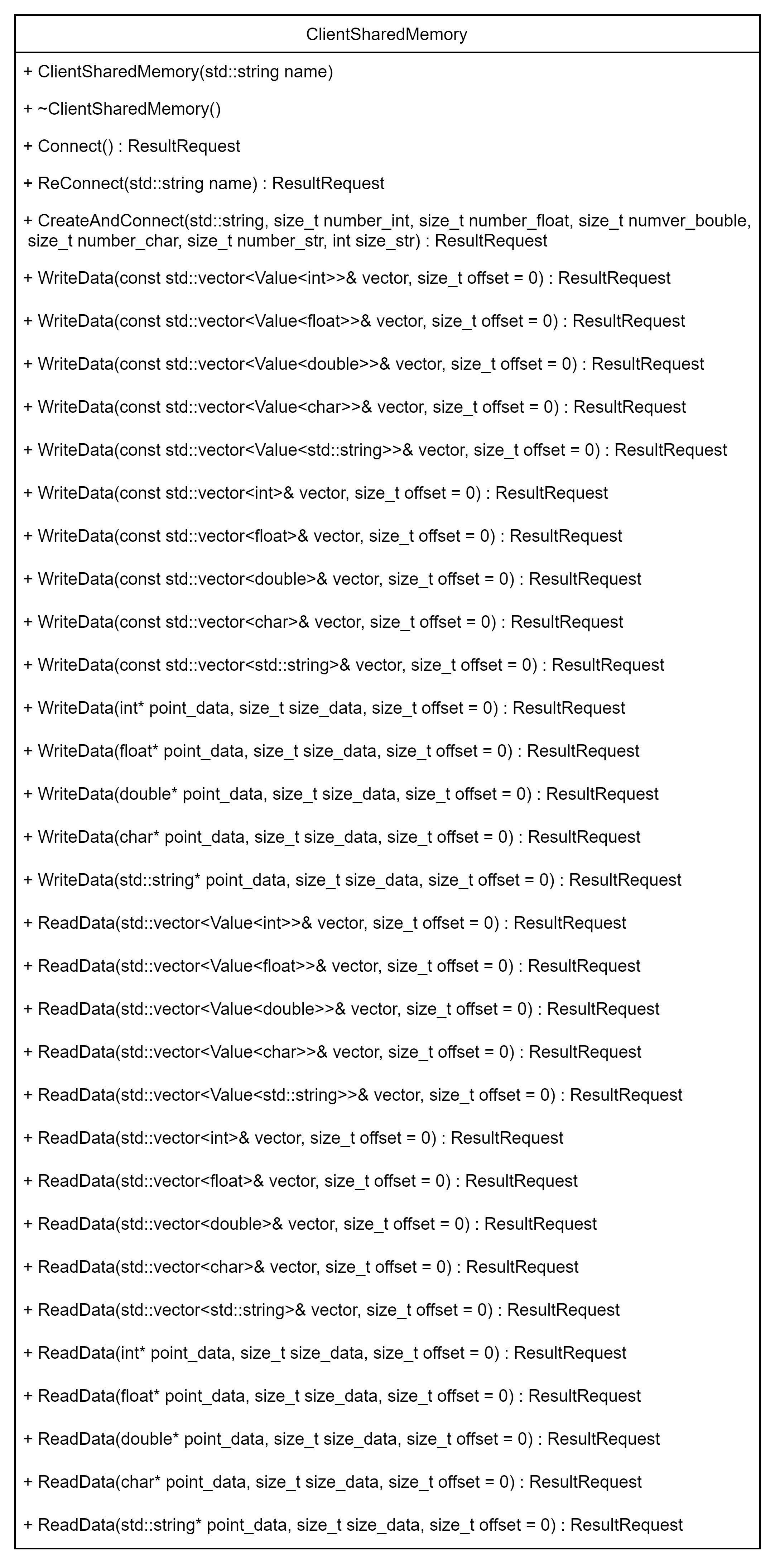


Рисунок 9 – Диаграмма класа ClientSharedMemory.

В таблице Таблица 1 представлено описание методов класса «ClientSharedMemory».

Таблица 1 – Таблица методов класса «ClientSharedMemory».

|  |  |
| --- | --- |
| *Метод* | |
| ClientSharedMemory(std::string name) | |
| *Назначение* | |
| Конструктор класса ClientSharedMemory | |
| *Аргументы* | *Назначение аргументов* |
| name | Наименование объекта разделяемой памяти c которым будем взаимодействовать экземпляр класса |
| Результат | Назначение результата |
| - | - |
| *Метод* | |
| ~ClientSharedMemory(std::string name) | |
| *Назначение* | |
| деструктор класса ClientSharedMemory | |
| *Аргументы* | *Назначение аргументов* |
| - | - |
| Результат | Назначение результата |
| - | - |
| *Метод* | |
| ResultRequest Connect() | |
| *Назначение* | |
| функция установления соединения с объектом разделяемой памяти | |
| *Аргументы* | *Назначение аргументов* |
| - | - |
| Результат | Назначение результата |
| ResultRequest::GOOD | Успешно установлено соединение с объектом разделяемой памяти с именем name |
| ResultRequest::WAIT\_INITIALIZATION | Попытка вызвать метод повторно до окончания завершения предыдущего вызова |
| ResultRequest::ERROR\_INITIALIZATION\_SECURITYATTRIBUTS | Ошибка инициализации атрибутов безопасности. Для уточнение смотрите системны код ошибка. |
| ResultRequest::ERROR\_INITIALIZATION\_MUTEX | Ошибка инициализации мьютекса WIN API. Для уточнение смотрите системны код ошибка. |
| ResultRequest::ERROR\_OPENING\_FILEMAPPING | Ошибка открытия разделяемой памяти. Для уточнение смотрите системны код ошибка. |
| ResultRequest::ERROR\_MAPPING\_MEMORY | Ошибка мапирования разделяемой памяти. Для уточнение смотрите системны код ошибка. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| *Метод* | |
| ResultRequest ReConnect() | |
| *Назначение* | |
| функция разрывает соединение с предыдущем объектом разделяемой памяти (если оно было установлено) и инициализирует новое соединение с объектом разделяемой памяти с наименованием «name» | |
| *Аргументы* | *Назначение аргументов* |
| name | Наименование объекта разделяемой памяти c которым будем взаимодействовать экземпляр класса |
| Результат | Назначение результата |
| ResultRequest::GOOD | Успешно установлено соединение с объектом разделяемой памяти с наименованием name |
| ResultRequest::WAIT\_INITIALIZATION | Попытка вызвать метод повторно до окончания завершения предыдущего вызова |
| ResultRequest::ERROR\_INITIALIZATION\_SECURITYATTRIBUTS | Ошибка инициализации атрибутов безопасности. Для уточнения смотрите системны код ошибка. |
| ResultRequest::ERROR\_INITIALIZATION\_MUTEX | Ошибка инициализации мьютекса WIN API. Для уточнения смотрите системны код ошибка. |
| ResultRequest::ERROR\_OPENING\_FILEMAPPING | Ошибка открытия разделяемой памяти. Для уточнения смотрите системны код ошибка. |
| ResultRequest::ERROR\_MAPPING\_MEMORY | Ошибка мапирования разделяемой памяти. Для уточнения смотрите системны код ошибка. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| *Метод* | |
| ResultRequest CreateAndConnect(std::string name, size\_t number\_int = 0, size\_t num‑ber\_float = 0, size\_t number\_double = 0, size\_t number\_char = 0, size\_t number\_str = 0, size\_t size\_str = 0) | |
| *Назначение* | |
| функция создания и установления соединения с объектом разделяемой памяти (используется в случаи отсутствия шлюза для проведения отладочных работ) | |
| *Аргументы* | *Назначение аргументов* |
| name | Наименование объекта разделяемой памяти c которым будем взаимодействовать экземпляр класса |
| number\_int | количество данных типа int, которые хранит в себе общая память |
| number\_float | количество данных типа float, которые хранит в себе общая память |
| number\_double | количество данных типа double, которые хранит в себе общая память |
| number\_char | количество данных типа char, которые хранит в себе общая память |
| number\_str | количество данных типа string, которые хранит в себе общая память |
| size\_str | максимальный хранимый размер данных типа string |
| Результат | Назначение результата |
| ResultRequest::GOOD | Успешно создан и установлено соединение с объектом разделяемой памяти с именем name |
| ResultRequest::ERROR\_MEMORY\_ALREADY\_INITIALIZATED | Общая память с именем name уже существует и с ним установлено соединение. |
| ResultRequest::WAIT\_INITIALIZATION | Попытка вызвать метод соединения повторно до окончания завершения предыдущего вызова |
| ResultRequest::ERROR\_INITIALIZATION\_SECURITYATTRIBUTS | Ошибка инициализации атрибутов безопасности. Для уточнения смотрите системны код ошибка. |
| ResultRequest::ERROR\_INITIALIZATION\_MUTEX | Ошибка инициализации мьютекса WIN API. Для уточнения смотрите системны код ошибка. |
| ResultRequest::ERROR\_MEMORY\_ALREADY\_EXIST | Ошибка создания объекта разделяемой памяти. Объект разделяемой памяти с таким именем уже существует. |
| ResultRequest::ERROR\_CREATE\_FILEMAPPING | Ошибка создания объекта разделяемой памяти. Для уточнения смотрите системны код ошибка. |
| ResultRequest::ERROR\_MAPPING\_MEMORY | Ошибка мапирования разделяемой памяти. Для уточнения смотрите системный код ошибка. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| *Метод* | |
| Status GetStatus() | |
| *Назначение* | |
| возвращает текущий статус объекта типа ClientSharedMemory | |
| *Аргументы* | *Назначение аргументов* |
| - | - |
| Результат | Назначение результата |
| Status::Null | Объект не имеет соединения с разделяемой памятью |
| Status::GOOD | Объект имеет установленное соединение с разделяемой памятью |
| Status::INITIALIZATION | Объект находится на этапе инициализации соединения с разделяемой памятью |
| Status::ERROR\_CONNECTION | Предыдущая попытка установления соединения с разделяемой памятью была неуспешной. |
| *Метод* | |
| DWORD GetSysError() | |
| *Назначение* | |
| Возвращает код последней системной ошибки WIN API | |
| *Аргументы* | *Назначение аргументов* |
| - | - |
| Результат | Назначение результата |
| (значение типа DWORD) | Код ошибки WIN API |
| *Метод* | |
| ResultRequest WriteData( const std::vector<Value<int>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа Value<int> | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<Value<float>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа Value<float> | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<Value<double>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа Value<double> | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<Value<char>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа Value<char> | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<Value<std::string>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа Value<std::string> | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<int>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа int | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<float>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа float | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<double>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа double | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<char>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа char | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData( const std::vector<std::string>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа string | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData(int\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа int | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | указатель на массив с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| size\_data | количество данных в массиве point\_data |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData(float\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа float | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | указатель на массив с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| size\_data | количество данных в массиве point\_data |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData(double\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа double | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | указатель на массив с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| size\_data | количество данных в массиве point\_data |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData(char\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа char | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | указатель на массив с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| size\_data | количество данных в массиве point\_data |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest WriteData(std::string\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция записи в объект разделяемой памяти данных типа std::string | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | указатель на массив с данными, которые необходима записать в объект разделяемой памяти (при этим качество данных устанавливается равным 0). |
| size\_data | количество данных в массиве point\_data |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается запись. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<Value<int>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа Value<int> (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Для уточнения смотрите системный код ошибкиСмещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<Value<float>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа Value<float> (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Для уточнения смотрите системный код ошибкиСмещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<Value<double>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа Value<double> (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<Value<char>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа Value<char> (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Для уточнения смотрите системный код ошибкиСмещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<Value<std::string>>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа Value<std::string> (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Для уточнения смотрите системный код ошибкиСмещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<int>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа int (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Для уточнения смотрите системный код ошибкиСмещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<float>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа float (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<double>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа double (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<char>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа char (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::vector<std::string>& vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа string (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(int\* point\_data, size\_t size\_ vector, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения из объект разделяемой памяти данных и их запись в vector типа int (количество считанных данных соответствует размеру вектор vector) | |
| *Аргументы* | *Назначение аргументов* |
| vector | Ссылка на вектор, в который необходима записать данные из объект разделяемой памяти |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(int\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения данных типа int из объект разделяемой памяти | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | указатель на массив с данными, в который необходима записать данные из разделяемой памяти |
| size\_data | количество данных в массиве point\_data (соответствует количеству считываемых данных) |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(float\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения данных типа float из объект разделяемой памяти | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | Указатель на массив с данными, в который необходима записать данные из разделяемой памяти |
| size\_data | Количество данных в массиве point\_data (соответствует количеству считываемых данных) |
| offset | Для уточнения смотрите системный код ошибкиСмещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(double\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения данных типа double из объект разделяемой памяти | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | Указатель на массив с данными, в который необходима записать данные из разделяемой памяти |
| size\_data | Количество данных в массиве point\_data (соответствует количеству считываемых данных) |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(char\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения данных типа char из объект разделяемой памяти | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | Указатель на массив с данными, в который необходима записать данные из разделяемой памяти |
| size\_data | Количество данных в массиве point\_data (соответствует количеству считываемых данных) |
| offset | Смещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |
| *Метод* | |
| ResultRequest ReadData(std::string\* point\_data, size\_t size\_data, size\_t offset = 0) | |
| *Назначение* | |
| функция чтения данных типа string из объект разделяемой памяти | |
| *Аргументы* | *Назначение аргументов* |
| point\_data | Указатель на массив с данными, в который необходима записать данные из разделяемой памяти |
| size\_data | Количество данных в массиве point\_data (соответствует количеству считываемых данных) |
| offset | Для уточнения смотрите системный код ошибкиСмещение в последовательности данных соответствующего типа, откуда начинается чтение. |
| Результат | Назначение результата |
| ResultRequest::GOOD; | Запрос выполнен успешно. |
| ResultRequest::ERROR\_BLOCK\_MUTEX; | Ошибка блокировки мьютекса. Для уточнения смотрите системный код ошибки. |
| ResultRequest::ERROR\_EXCESS\_OFFSET | Смещение превышает количество данных, хранящихся в разделяемой памяти. |
| ResultRequest::ERROR\_NOT\_INDIFICATED | Ошибка не идентифицирована. |
| ResultRequest::WARNANING\_EXCESS\_RANGE; | Предупреждение. Количество данных в запросе превышает количество, которое хранится в памяти. |

# Запуск и выполнение программы

Для успешного соединения с шлюзом необходимо выполнить его запуск до выполнения ПО ЭМТ (иначе необходимо периодические выполнять попытку соединения при помощи библиотеки «ClientSharedMemory» до момента успешного выполнения метода).

Для запуска шлюза необходимо включить узел ЭМТ и войти в операционную систему.

После загрузки ОС приложение автоматически запускается и начинает чтение конфигурации и ее применение.

В случае возникновения ошибки в ходе запуска приложение будет автоматически перезапускаться. В случае сбоя сообщение об ошибке будет записано на локальный диск в файл журнала.

При разрыве TCP/IP соединения шлюз будет осуществлять повторное подключение. Информация об этом выводится в локальный файл журнала.

# Сообщения системному программисту

При запуске и работе программы в локальный файл журнала выводятся сообщения о отладке, информационные, предупреждения и ошибки. В зависимости от настроек уровня журналирования. Для записи сообщений в локальный журнал используется файл кодировки UTF8 на локальный диск в каталог, где размещено программное обеспечение. Название файла имеет формат “LogGate.txt” и включает в себя год, месяц и день. Файл содержит следующую информацию:

* дата и время события;
* статус события (DEBUG – отладка, INFO - информационное, WARNING - предупреждение, ERROR - ошибка);
* локальный код ошибки;
* системный код ошибки операционной системы;
* текстовое описание события на английском языке.

В таблице 2 приведен список возможных сообщений.

Таблица 2 – Перечень сообщений системному программисту

| № п/п | Сообщение | Локальный код ошибки | Статус сообщения | Причина возникновения | Действия системного программиста |
| --- | --- | --- | --- | --- | --- |
|  | ERROR CONFIG\_FILE OF GATE | 1 | WARNING | Ошибка открытия конфигурационного файла config.json | Создать файл конфигурации и разместить его в одной директории со шлюзом |
|  | ERROR CONFIG\_FILE OF GATE | 1 | WARNING | Отсутствует заголовок конфигурации GATE | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIG\_FILE OF LOGGER | 1 | WARNING | Ошибка в конфигурации  LOGGER | Исправить файл конфигурации шлюза шлюза config.json |
|  | ERROR CONFIG\_FILE OF MANAGER | 1 | WARNING | Ошибка в конфигурации  MANAGER | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIGUNITS\_FILE | 1 | WARNING | Ошибка открытия файла configunits.json | Создать файл конфигурации и разместить его в одной директории со шлюзом |
|  | ERROR CONFIG\_FILE OF LOGGER | 2 | WARNING | Ошибка в конфигурации LogMode | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIG\_FILE OF LOGGER | 3 | WARNING | Ошибка в конфигурации StatusLog | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIG\_FILE OF LOGGER | 4 | WARNING | Ошибка в конфигурации StatusSysLog | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIG\_FILE OF LOGGER | 5 | WARNING | Ошибка в конфигурации SizeLogFile | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIG\_FILE OF LOGGER | 6 | WARNING | Ошибка в конфигурации LogName | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIG\_FILE OF LOGGER | 7 | WARNING | Ошибка в конфигурации SysLogName | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIG\_FILE OF MANAGER | 2 | WARNING | Ошибка в конфигурации IP адресе наблюдателя | Исправить файл конфигурации шлюза config.json |
|  | ERROR CONFIG\_FILE OF MANAGER | 3 | WARNING | Ошибка в конфигурации Port наблюдателя | Исправить файл конфигурации шлюза |
|  | ERROR INITIALIZATION OF SECURITY\_ATTRIBUTES | 1 | DEBUG | Ошибка в функции AllocateAndInitializeSid | Сообщение для разработчиков |
|  | ERROR INITIALIZATION OF SECURITY\_ATTRIBUTES | 2 | DEBUG | Ошибка в функции SetEntriesInAclA | Сообщение для разработчиков |
|  | ERROR INITIALIZATION OF SECURITY\_ATTRIBUTES | 3 | DEBUG | Ошибка в функции LocalAllc | Сообщение для разработчиков |
|  | ERROR INITIALIZATION OF SECURITY\_ATTRIBUTES | 4 | DEBUG | Ошибка в функции InitializeSecurityDescriptor | Сообщение для разработчиков |
|  | ERROR INITIALIZATION OF SECURITY\_ATTRIBUTES | 5 | DEBUG | Ошибка в функции SetSecurityDescriptorDacl | Сообщение для разработчиков |
|  | ERROR INITIALIZATION GATE | 1 | ERROR | Ошибка инициализации шлюза | Предпринимать действия в соответствии системному коду ошибки |
|  | START GATE | NULL | INFO | Инициализирован  шлюз | Не требуются |
|  | ERROR MANAGER | 1 | DEBUG | Ошибка работы менеджера по в функции WSAStartup | Сообщение для разработчиков |
|  | ERROR MANAGER | 2 | DEBUG | Ошибка работы менеджера по в функции WSASocket | Сообщение для разработчиков |
|  | ERROR MANAGER | 3 | DEBUG | Ошибка работы менеджера по в функции функции bind | Сообщение для разработчиков |
|  | ERROR WRITE OF SHARED MEMORY | 1 | DEBUG | Не инициализирован мьютекс для записи для разделяемой памяти | Сообщение для разработчиков |
|  | ERROR WRITE OF SHARED MEMORY | 2 | DEBUG | Не инициализирован память для записи в разделяемой памяти | Сообщение для разработчиков |
|  | ERROR WRITE SHARED MEMORY | 4 | DEBUG | Не инициализирован буфер для записи в разделяемую память | Сообщение для разработчиков |
|  | ERROR READ OF SHARED MEMORY | 1 | DEBUG | Не инициализирован мьютекс для чтения разделяемой памяти | Сообщение для разработчиков |
|  | ERROR READ OF SHARED MEMORY | 2 | DEBUG | Не инициализирована память для чтения в разделяемой памяти | Сообщение для разработчиков |
|  | ERROR READ SHARED MEMORY | 4 | DEBUG | Не инициализирован буфер для чтения разделяемой памяти | Сообщение для разработчиков |
|  | ERROR INITIALIZATION OF COMMAND-MEMORY | 4 | DEBUG | Ошибка отображения памяти команд | Сообщение для разработчиков |
|  | ERROR DROP COMMAND FOR SHARED MEMORY | 1 | DEBUG | Ошибка сброса команды для обшей памяти. Не инициализирован мьютекс командной памяти | Сообщение для разработчиков |
|  | ERROR DROP COMMAND FOR SHARED MEMORY | 2 | DEBUG | Ошибка сброса команды для обшей памяти. Не инициализирована командная память | Сообщение для разработчиков |
|  | ERROR WRITING COMMAND FOR EMT | 1 | DEBUG | Ошибка записи команды обновления данных для ЭМТ. Не инициализирован мьютекс командной памяти | Сообщение для разработчиков |
|  | ERROR WRITING COMMAND FOR EMT | 2 | DEBUG | Ошибка записи команды обновления данных для ЭМТ. Не инициализирована командная память | Сообщение для разработчиков |
|  | ERROR RECEIVING REQUEST FROM MANAGER | NULL | DEBUG | Ошибка приема запроса от менеджера шлюза | Сообщение для разработчиков |
|  | ERROR TRANSFER A RESPONSE TO MANAGER | NULL | DEBUG | Ошибка передачи ответа на запрос менеджера шлюза | Сообщение для разработчиков |
|  | ERROR INITAILIZATION SOCKET TO CONTROL | NULL | ERROR | Ошибка инициализации сокета для управления шлюзом | Предпринимать действия в соответствии системному коду ошибки |
|  | START SOCKET TO CONTROL IP: \*IP PORT: \*Port | NULL | INFO | Сообщение об успешном запуске сокета для управления шлюзом | Не требуются |
|  | STOPPING GATE ON COMMAND | NULL | INFO | Остановка работы шлюза по команде | Не требуются |
|  | COMMAND NOT DEFINED | NULL | ERROR | Ошибка. Не определена запрашиваемая команда | Проверить корректность запрашиваемой команды |
|  | ERROR INTIALIZATION OF SOCKET \*CodeError | NULL | ERROR | Ошибка инициализации сокета (\*CodeError - системный код ошибки) | Предпринимать действия в соответствии с системным кодом ошибки |
|  | ERROR CONNECTING \*CodeError | NULL | ERROR | Ошибка соединения с шлюзом (\*CodeError - системный код ошибки) | Предпринимать действия в соответствии с системным кодом ошибки |
|  | ERROR RECEIVE PACKET OF ANSWER \*CodeError | NULL | ERROR | Ошибка приема ответа на запрос (\*CodeError - системный код ошибки) | Предпринимать действия в соответствии с системным кодом ошибки |
|  | ERROR TRANSFER PACKET OF REQUEST \*CodeError | NULL | ERROR | Ошибка отправки запроса шлюзу (\*CodeError - системный код ошибки) | Предпринимать действия в соответствии с системным кодом ошибки |
|  | ERROR REQUEST: COMMAND INACCESSIBLE | NULL | ERROR | Ошибка запроса. Функция не поддерживается данным шлюзом | Проверить корректность запрашиваемой команды |
|  | ERROR REQUEST: NOT DONE | NULL | ERROR | Ошибка запроса. Устройство вернуло отрицательный ответ на запрос | Уточнить ошибку в конфигурационных файлах устройства, к которому адресована команда |
|  | ERROR GETTING STATUS | NULL | ERROR | Ответ на команду getstatusgate. Ошибка запроса | Не требуются |
|  | СURRENT STATUS OF GATE: \*STATUS | NULL | INFO | Ответ на команду getstatusgate. \*STATUS - текущий статус | Не требуются |
|  | ERROR STARTING OF GATE | NULL | ERROR | Ответ на команду startgate. Ошибка запуску шлюза | Не требуются |
|  | GATE STARTED | NULL | INFO | Ответ на команду startgate. Шлюз запущен | Не требуются |
|  | ERROR STOPING OF GATE | NULL | ERROR | Ответ на команду stopgate. Ошибка остановки шлюза | Не требуются |
|  | GATE STOPED | NULL | INFO | Ответ на команду stopgate. Шлюз остановлен | Не требуются |
|  | ERROR GETTING STATUS | NULL | ERROR | Ответ на команду getstatus. Ошибка запроса | Не требуются |
|  | INVALID ARGUMENT: INSUFFICIENT INPUT PARAMETRS | NULL | ERROR | Ошибка запроса. Недостаточно входных параметров | Проверить корректность запрашиваемой команды |
|  | ERROR TRANSFER PACKET TO SCADA | 1 | DEBUG | Ошибка передачи пакета. Ошибка в функции write. | Сообщение для разработчиков |
|  | ERROR TRANSFER PACKET TO SCADA | 2 | DEBUG | Ошибка передачи пакета. Ошибка в функции wait | Сообщение для разработчиков |
|  | ERROR TRANSFER PACKET TO SCADA | 4 | DEBUG | Ошибка передачи пакета шлюзу. Ошибка в функции aio\_return | Сообщение для разработчиков |
|  | ERROR RECEIVE PACKET FROM SCADA | 1 | DEBUG | Ошибка приема данных от SCADA. Ошибка в функции read | Сообщение для разработчиков |
|  | ERROR RECEIVE PACKET FROM SCADA | 2 | DEBUG | Ошибка приема данных от SCADA. Ошибка в функции aio\_suspend | Сообщение для разработчиков |
|  | ERROR RECEIVE PACKET FROM SCADA | 4 | DEBUG | Ошибка приема данных от SCADA. Ошибка в функции aio\_return | Сообщение для разработчиков |
|  | TRANSFER PROCESS TOOK PLACE AFTER \*time ms | NULL | INFO | Предупреждение. Запрос на чтение данных произошел после \*time ms, относительно предыдущего запроса | Не требуются |
|  | ERROR LOCK BLOCK FILE | 1 | DEBUG | Ошибка блокировки блокирующего файла. Ошибка в функции open | Сообщение для разработчиков |
|  | ERROR LOCK BLOCK FILE | 2 | DEBUG | Ошибка блокировки блокирующего файла. Ошибка в функции flock | Сообщение для разработчиков |

# Перечень сокращений

|  |  |
| --- | --- |
| АРМ | Автоматизированное рабочее место |
| АТЭ | Атомтехэнерго |
| ОС | Операционная система |
| ПНР | Пусконаладочные работы |
| ПО | Программное обеспечение |
| РАЭК | Расчетно-аналитический (экспериментальный) комплекс |
| ТПТС | Программно-технические средства производства ФГУП «ВНИИА» |
| ЭМТ | Эмулятор ТПТС |
| SCADA | Supervisory Control And Data Acquisition – пер. с англ. Диспетчерское управление и сбор данных |

# Лист изменений (дополнений)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата | Номер изменения | Номер измененного (дополненного) пункта/раздела | Изменение внес | |
| ФИО | Подпись |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |