

Спецификация открытого протокола обмена
NordWind Open UDP
(Версия 2.6)

Руководитель разработки
Технический директор

_____ Баум Ф.И.
«_____» _____ 2026 г.

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ОПИСАНИЕ ПРИНЦИПОВ РАБОТЫ ПРОТОКОЛА	3
1.1 Обмен данными с предварительно созданным каналом	3
1.2 Обмен данными без создания канала обмена данными.....	6
2 ОПИСАНИЕ ФУНКЦИЙ ПРОТОКОЛА ИСПОЛЬЗУЮЩИХ КАНАЛ ОБМЕНА ДАННЫМИ	7
3 ОПИСАНИЕ ФУНКЦИЙ ПРОТОКОЛА НА ОСНОВЕ РАСПОЛАГАЕМОГО СПИСКА СИГНАЛОВ	13
4 АДАПТАЦИЯ БИБЛИОТЕКИ ПРОТОКОЛА К ПРОПУСКНЫМ ВОЗМОЖНОСТЯМ КАНАЛА СВЯЗИ	22

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

ВВЕДЕНИЕ

Программное обеспечение NordWind использует универсальный открытый протокол обмена данными NordWind Open UDP для передачи информации между своими компонентами и вычислительными узлами. Этот протокол основан на стеке UDP/IP и отвечает ряду важных требований:

1. Поддержка передачи различных типов данных, таких как числа с плавающей запятой и целые числа (double, int, unsigned char), статусов сигналов и времен их возникновения.
2. Обеспечение целостности данных при сетевом обмене.
3. Возможность диагностики ошибок и потерь данных.
4. Совместим с разным типом ОС и архитектур процессоров.
5. Адаптация скорости обмена данными к характеристикам линии передачи для уменьшения потерь.
6. Две группы функций: для высокоскоростной сети (с созданием канала обмена данными) и для низкоскоростной сети (без канала обмена данными).
7. Реализация клиентской части на языке Си с открытым исходным кодом;

Протокол предназначен для обмена данными в автоматизированных системах управления технологическими процессами (АСУ ТП). Разработчики могут модифицировать функции протокола в зависимости от своих задач.

Протокол NordWind Open UDP является универсальным и может применяться не только с программами SimInTech и NordWind, но также с другими аппаратными и программными платформами.

1 ОПИСАНИЕ ПРИНЦИПОВ РАБОТЫ ПРОТОКОЛА

1.1 Обмен данными с предварительно созданным каналом

Если скорость сети составляет 100 кбит/с или выше, рекомендуется использовать функции с предварительно созданным каналом обмена. Канал представляет собой структуру, создаваемую на стороне клиента и сервера для описания сигналов, между которыми будет происходить обмен данными. Этот процесс называется «Сессия обмена данными».

Для организации канала необходимо создать структуру «GROUP», которая упорядочивает данные, полученные от программного обеспечения NordWind, для дальнейшей работы с ними.

Содержимое структуры «GROUP» приведено ниже см. (Рисунок 1).

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

```

struct GROUP
{
    int            *indx; /*Массив индексов сигналов*/
    int            *len;  /*Размер значения сигнала*/
    unsigned char  *data_type; /*Тип данных 0-вещественный, 1-двоичный, 2-целый*/
    int            num;   /*Количество сигналов в группе*/
    int            size;  /*Размер значений для группы сигналов*/
    unsigned char  *value; /*Хранятся значения в байтовом потоке*/
    int            *status; /*Массив статусов сигнала*/
    double         *time;  /*Массив времен возникновения сигналов*/
    int            *offset; /*Хранятся смещения*/

};

```

Рисунок 1 – Структура данных в структуре «GROUP»

Эта структура помогает управлять данными и оптимизировать обмен между клиентом и сервером.

Структура «GROUP» (группа) хранит значения, статусы, времена возникновения сигналов, а также информацию для выполнения операций с данными группы.

Протокол подразумевает использование двух групп данных:

- 1) группа на стороне клиента (удаленная группа);
- 2) группа на стороне сервера обмена NordWind_UDP_Server (локальная группа).

Локальная группа получает данные из памяти (базы данных) программного обеспечения NordWind.

Под обменом данных подразумевается обмен данными между локальной и удаленной группами (удаленная группа является проекцией локальной группы). Для работы с группой существуют функции, которые позволяют оперировать данными из нее: через использование индекса может происходить получение или запись соответствующих данных. Таким образом, операции получения и записи данных в память NordWind выполняются через структуру «GROUP».

Для обмена данными между клиентом и базой данных NordWind необходим сервер NordWind_UDP_Server. Этот сервер работает как UDP-сервер и имеет доступ к разделяемой памяти NordWind (в ней хранится база сигналов).

Для использования протокола обмена данными необходимо:

1. Инициализировать соединение с сервером на стороне клиента.
2. Открыть «Сессию обмена данными», создавая канал между клиентом и сервером. При успешном создании структуры «GROUP» на обеих сторонах обмен становится возможным. Если возникла ошибка, необходимо повторить команду создания канала. Эта команда выполняется один раз, и нужно создать канал заново в случае перезапуска сервера.
3. После открытия «Сессии обмена данными» доступны функции протокола для обмена. Важно проверять код возврата функций: 0x001 означает успешное выполнение, отрицательные значения указывают на ошибки, такие как потеря связи

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

или несоответствие контрольных сумм. Если произошла ошибка, необходимо выяснить ее причину и, в случае отсутствия канала, создать его заново.

4. Сервер может обрабатывать команды от нескольких клиентов, которые открыли свои «Сессии обмена данными».
5. «Сессии обмена данными» завершена если от клиента получена команда на закрытие в результате на стороне сервера будет освобождена память от структуры со списком сигналов обмена «GROUP» и для продолжения обмена необходимо заново открыть «Сессию обмена данными».

На стороне сервера для каждого клиента создается структура данных канала (информация необходимая для управления обменом) и заносится в хэш-очередь по IP-адресу. Для каждого клиента (при этом подразумевается, что каждый клиент имеет свой IP-адрес) создается только одна структура данных канала (в состав этой структуры включена «GROUP»).

При отправке команд со стороны клиента сервер сначала производит проверку того, создана ли для этого клиента структура данных канала или нет, а затем по результатам проверки возможны следующие варианты:

- 1) если группа существует, то команда будет выполнена;
- 2) если группа не существует, то будет выдан код ошибки.

При использовании канала обмена данными обычно доступны шесть команд, от `cmd=0` до `cmd=5`. Первое обращение клиента должно быть с командой `cmd=1`, которая создает канал обмена. При успешной реализации команд клиент получит необходимые данные или код возврата. Если команда выполнена успешно, код возврата будет положительным; в случае ошибок он будет отрицательным.

Протокол определяет несколько ошибок:

- отсутствие запрашиваемого сигнала при создании канала обмена;
- таймаут;
- неверные контрольные суммы;
- отсутствие созданной «Сессии обмена данными» для клиента;
- ошибка передачи по сети.

Важно отметить, что при обмене данными между клиентом и сервером размер буфера данных может превышать 1400 Байт, поэтому при посылке данных в сеть по протоколу UDP буфер данных будет разбит на отдельные пакеты размером 1400 Байт, таким образом мы гарантируем отсутствие фрагментации пакета и можем быть уверены в целостности данных внутри каждого пакета (протокол UDP использует анализ контрольной суммы при приеме/передаче пакетов с данными по сети). При передаче данных по UDP возможны потери пакетов, для оценки целостности данных используется расчет контрольной суммы CRC32 для всего передаваемого/принимаемого буфера данных, таким образом после завершения приема/передачи можно оценить целостность данных и в случае отсутствия таковой повторить обмен.

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

1.2 Обмен данными без создания канала обмена данными

Данную группу функций целесообразно использовать при условии существования сети с низкой пропускной способностью (~10 Кбод) и отсутствие необходимости получать значения статусов и времен возникновения сигналов. Предварительно необходимо подготовить три файла («discrets.dat», «analog.s.dat», «int.dat») со списками имен дискретных, аналоговых и целочисленных сигналов соответственно, именно с сигналами из перечисленных файлов будет происходить обмен данными. При реализации обмена данными с использованием описываемой группы функций данные передаются в составе одного пакета по протоколу UDP размер которого не превышает 1400 Байт, дефрагментации пакета не происходит, а использование контрольной суммы протоколом UDP позволяет быть уверенным в целостности данных. Адресация данных будет определяться порядком нахождения имен сигналов в соответствующих файлах. Вне зависимости от расположения сигналов в базе данных NordWind. Файлы со списком сигналов должны быть расположены в рабочей директории сервера NordWind_UDP_Server, при старте сервера если хоть один из сигналов не будет найден в базе данных NordWind сервер прекращает свою работу с соответствующей диагностикой. Для использования протокола необходимо:

1. Со стороны клиента инициализировать соединение с сервером.
2. При использовании функций протокола анализировать коды возврата и в случае отрицательных кодов возврата использовать функции повторно.
3. В случае перезапуска сервера нет необходимости в повторном переинициализации соединения

Протокол выполняет следующее определение ошибок:

- Потерю пакетов с данными;
- Неверную структуру данных;
- Ошибки сети;

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

2 ОПИСАНИЕ ФУНКЦИЙ ПРОТОКОЛА ИСПОЛЬЗУЮЩИХ КАНАЛ ОБМЕНА ДАННЫМИ

В протоколе NordWind Open UDP предусмотрены функции для работы с данными через предварительно созданный канал обмена. В этом разделе перечислены функции, которые обеспечивают обмен данными между удаленной и локальной группами на клиенте и сервере соответственно. При передаче данных по сети пакеты будут содержать заголовок для идентификации и обработки.

```

/*Заголовок пакета для формирования группы*/
typedef struct
{
    uint8_t      cmd;                /*Команда*/
    uint32_t     nbytes;             /*Размер пакета*/
    uint32_t     num_pakets_gr;      /*Общее количество пакетов для
формирования группы*/
    uint32_t     indx0;              /*Начальный индекс сигнала в группе*/
    uint32_t     nval;               /*Количество сигналов включая индекс*/
    uint32_t     num_signals_gr;     /*Количество сигналов в группе*/
    uint32_t     num_signals_in_packet; /*Количество имен в пакете*/
    uint32_t     num_current;        /*Конкретный номер пакета*/
    uint32_t     crc32;              /*Контрольная сумма всех передаваемых
данных*/
} Header;

```

Функции находятся в файле «nw_exchange.c», а их заголовки — в файле «nw_exchange.h». Эти файлы предоставляются разработчиком протокола вместе с остальными материалами, по запросу.

1.

```

int Init_NW_Exchange_Udp_Channel(Channel_Data *channel_data, char *ip_addr,
uint16_t port)

```

Обмен данными происходит с использованием протокола UDP.

В результате вызова этой функции создается присоединенный сокет, который необходим для обеспечения постоянного адреса отправителя в пакете. Если сокет не присоединен, ip адрес пакета с данными может изменяться, что затрудняет идентификацию. Благодаря присоединенному сокету сервер может определить группу данных, к которой относится запрос от клиента, основываясь на указанном ip адресе клиента. Происходит выделение памяти для структуры «GROUP».

2.

```
int Create_an_Exchange_Channel(Channel_Data *channel_data, char (*name)[64],
uint32_t num )
```

При использовании данной функции на стороне клиента создается структура данных «GROUP», после чего аналогичная структура создается на сервере, а также формируются необходимые структуры данных для обмена по данному IP-адресу. Если функция вызывается повторно, и группа уже существует на сервере, возвращается код ошибки.

В результате выполнения функции на стороне клиента и сервера будет получена инициализированная группа для определенного списка сигналов.

Обмен данными между клиентом и сервером приведен в таблицах 1 и 2.

Таблица 1 – Создание локальной группы на стороне сервера и данные, посылаемые клиентом пакеты не больше 1400 Байт, структура пакета

Описание	Размер	Значения
Заголовок Header: {cmd=0x01,...}	33 Байт	Cmd=0x01
Массив пар {len, name} [Num_Signal_Packet], где Num_Signal_Packet количество сигналов в пакете		
Длина имени сигнала len	uint32_t	Целое число
Имя сигнала name	char[64]	Массив байт

Таблица 2 – Ответ сервера на запрос клиента, пакеты размер не больше 1400 Байт

Описание	Размер	Значения
Заголовок Header: {cmd=0x01,...}	33 Байт	Cmd=0x01
Массив пар {len, data_type}[Num_Signal_Packet], где Num_Signal_Packet количество сигналов в пакете		
Массив значений размеров сигналов len	uint32_t размер элемента массива	Массив целых чисел. Если сигнала отсутствует в базе данных (shm) NW, то значение 1 Байт
Массив значений типов данных сигналов data_type	uint8_t размер элемента массива	Массив целых чисел. Если сигнала отсутствует в базе данных (shm) NW, то значение 255

Контрольная сумма `crc32`, указанная в заголовке, рассчитывается для всех пар `{len, name}`, составляющих канал обмена.

Коды ошибок функции:

- 1 группа создана;
- 1 ошибка сети, тайм-аут, канал уже создан;
- 2 нет сигнала в бд NW.
- 3 неверная контрольная сумма.

3.

`int Read_All_Data_From_Channel(Channel_Data *channel_data)`

Эта функция запрашивает значения всех сигналов на UDP-сервере. При синхронизации вы получите значения, статусы и время появления сигналов для всех сигналов в группе. Чтобы получить значения сигналов из этой группы, используйте функцию «`get_value_from_group()`» и укажите номер сигнала в группе.

Для обновления группы клиент отправляет серверу сообщение размером 9 байт: `{cmd = 0x03, indx0 = 0, nval = num (количество сигналов в канале)}`. Ответ от сервера можно найти в таблице 3.

Таблица 3 – Ответ сервера на запрос клиента `cmd=0x03`, пакеты размер не больше 1400 Байт

Описание	Размер	Значения
Заголовок Header: <code>{cmd=0x03,...}</code>	33 Байт	<code>Cmd=0x03</code>
Массив <code>{value, status, time}[Num_Signal_Packet]</code> , где <code>Num_Signal_Packet</code> количество сигналов в пакете	<code>uint8_t</code>	Массив байт

Контрольная сумма `crc32`, указанная в заголовке, рассчитывается для всех значений `{value, status, time}` списка сигналов.

Коды ошибок функции:

- 1 синхронизация произошла;
- 1 ошибка сети, тайм-аут;
- 3 не совпадение контрольной суммы.

4.

`int Destroy_the_Exchange_Channel(Channel_Data *channel_data)`

Эта функция завершает «Сессию обмена данными» и освобождает ресурсы на стороне клиента и сервера.

Для реализации данного функционала происходит отправка клиентом команды `cmd=0x02`.

Коды ошибок функции:

- 1 группа удалена;
- 1 ошибка сети, тайм-аут;
- 2 группа не создана для клиента;

5.

`int Write_All_Data_To_Channel(Channel_Data *channel_data)`

Обновляем значения списка сигналов, входящих в канал обмена (локальной группы) на стороне UDP-сервера, после чего данные значений сигналов, статусы, времена возникновения записываются в бд NW. Мы имеем возможность менять данные в группе на стороне клиента используя функцию «set_value_in_group()» перед использованием функции «Write_All_Data_To_Channel ()», но после вызова данной функции все данные группы будут переданы по сети серверу, даже если они не менялись! Формат данных, посланного клиентом серверу (Таблица 4).

Таблица 4 – Запрос клиента, пакеты размер не больше 1400 Байт

Описание	Размер	Значения
Заголовок Header: {cmd=0x04,...}	33 Байт	Cmd=0x04
Массив {value, status, time}[Num_Signal_Packet], где Num_Signal_Packet количество сигналов в пакете	uint8_t	Массив байт

Контрольная сумма crc32, приведенная в заголовке рассчитана для всех {value, status, time} сигналов входящих в канала обмена.

Коды ошибок функции:

- 1 данные записаны в бд NW;
- 1 ошибка сети, тайм-аут;
- 3 неверная контрольная сумма;

6.

`Read_Data_From_Channel(Channel_Data *channel_data, int indx0, int nval)`

Используя эту функцию, мы запрашиваем не все значения списка сигналов, входящих в канал обмена (группы), а только заданные (indx0 индекс в группе сигнала, nval количество сигналов начиная с indx0) и по сети будет передан пакет только с определенными нами сигналами. Перед формированием пакета UDP сервер обновит данные группы. Используя функцию «get_value_from_group ()» можно будет получить значения полученных сигналов. Формат данных, посланного клиентом серверу (Таблица 5), ответ сервера (Таблица 6).

Таблица 5 – Запрос клиента на чтение сигналов

Описание	Размер	Значения
----------	--------	----------

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
---	---	-----

Команда серверу на чтение сигналов из БД NW в группу cmd	1 Байт	0x03
Начальный индекс сигнала в группе indx0	uint32_t	Целое число
Количество считанных сигналов nval	uint32_t	Целое число

Таблица 6 – Ответ сервера, пакеты размер не больше 1400 Байт

Описание	Размер	Значения
Заголовок Header: {cmd=0x03,...}	33 Байт	Cmd=0x03
Массив {value, status, time}}[Num_Signal_Packet], где Num_Signal_Packet количество сигналов в пакете	uint8_t	Массив байт

Контрольная сумма crc32, приведенная в заголовке рассчитана для всех {value, status, time} выборки indx0, nval.

Коды ошибок функции:

- 1 обновление произошло;
- 1 ошибка сети, тайм-аут;
- 3 не совпадение контрольной суммы.

7.

```
int Write_Data_To_Channel(Channel_Data *channel_data, int indx0, int nval)
```

Используя эту функцию, мы выбираем из группы (на стороне клиента) сигналы (indx0, nval) и посылаем их серверу для того, чтобы он обновил их в локальной группе (на стороне сервера). После успешного обновления сигналов в локальной группе сервер записывает значения только обновленных сигналов из группы в бд NW, запись в бд NW происходит в том случае если не было потерь пакетов с данными по сети (для контроля целостности данных используется CRC32). Изменить сигналы в группе перед отправкой можно используя функцию «set_value_in_group ()». Формат данных, посланного клиентом серверу (Таблица 7).

Таблица 7 – клиент посылает пакеты для записи значений в локальную группу и потом в базе данных NordWind, размер пакетов не больше 1400 Байт.

Описание	Размер	Значения
Заголовок Header: {cmd=0x04,...}	33 Байт	Cmd=0x04
Массив {value, status, time}}[Num_Signal_Packet], где Num_Signal_Packet количество сигналов в пакете	uint8_t	Массив байт

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
---	---	-----

Контрольная сумма crc32, приведенная в заголовке рассчитана для всех {value, status, time} выборки indx0, nval.

Коды ошибок функции:

- 1 данные записаны в бд NW;
- 1 ошибка сети, тайм-аут;
- 3 неверная контрольная сумма;

8.

```
int Write2_Data_To_Channel(Channel_Data *channel_data, int indx0, int nval)
```

Используя эту функцию, мы выбираем из группы сигналы (indx0, nval) и посылаем их серверу для того, чтобы он обновил их в локальной группе (на стороне сервера). После доставки данные каждого пакета записываются в бд NW, контроль целостности данных по CRC32 не происходит. Использовать данную функцию полезно в том случае если посылка данных от клиента серверу происходит периодически через небольшой временной интервал. Изменить сигналы в группе перед отправкой можно используя функцию «set_value_in_group ()». Формат данных, посланного клиентом серверу (Таблица 7).

Таблица 7 – клиент посылает пакеты для записи значений в локальную группу и потом в базе данных NordWind, размер пакетов не больше 1400 Байт.

Описание	Размер	Значения
Заголовок Header: {cmd=0x04,...}	33 Байт	Cmd=0x04
Массив {value, status, time}[Num_Signal_Packet], где Num_Signal_Packet количество сигналов в пакете	uint8_t	Массив байт

Коды ошибок функции:

- 1 данные записаны в бд NW;
- 1 ошибка сети, тайм-аут;

9.

```
int Check_Channel(Channel_Data *channel_data)
```

Функция используется для проверки существования канала обмена данными для данного ip адреса. Запрос к серверу {(uint8_t)0x0, (uint16_t)0xFFFF, rand= (uint8_t) (rand() % 100)}, ответ сервера {(uint16_t)0xFFFF, rand, code}. Рекомендуется использовать эту функцию перед обменом данными с сервером.

Коды ошибок функции:

- 1 канал существует;
- 0 канал не существует;

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

- 1 ошибка сети, тайм-аут;
- 3 неверная структура ответа;

3 ОПИСАНИЕ ФУНКЦИЙ ПРОТОКОЛА НА ОСНОВЕ РАСПОЛАГАЕМОГО СПИСКА СИГНАЛОВ

Данную группу функций целесообразно использовать при условии существования сети с низкой пропускной способностью (~10 Кбод) и отсутствие необходимости получать значения статусов и времен возникновения сигналов. Данные для обмена должны быть представлены однотипными таблицами на стороне сервера. Для доступа к данным на стороне сервера необходимо задать начальный индекс однотипной таблицы и размер массива данных.

Для функций:

```
Read_BitsData_From_DB_NW ();
Write_BitsData_From_DB_NW ();
Read_IntData_From_DB_NW ();
Write_IntData_From_DB_NW ();
```

существует возможность изменять код команды cmd, это может быть удобно при разработке ПО обмена данными с ПО, использующим протокол modbus.

При реализации обмена данными с использованием описываемой группы функций данные передаются в составе одного пакета по протоколу UDP размер которого не превышает 1400 Байт, дефрагментации пакета не происходит, а использование контрольной суммы протоколом UDP позволяет быть уверенным в целостности данных

Функции, обеспечивающие контроль целостности данных, возвращают числовые коды состояния.

Коды разделены на две группы:

- **Отрицательные значения (-1, -2, -3)** – ошибки, возникающие на стороне клиента (сетевые проблемы или некорректные входные параметры, нарушения порядка приема пакетов).
- **Шестнадцатеричные коды (0x000...0x005)** – ошибки, возвращаемые серверной частью (проблемы с таблицами, адресацией или базой данных).

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
---	---	-----

Коды ошибок клиентской части (сетевые и валидационные)

Код	Название	Описание
-1	NET_TIMEOUT	Ошибка сети: превышен тайм-аут ожидания ответа от сервера.
-2	SIGNAL_COUNT_EXCEEDED	Размер запрашиваемого вектора сигналов превышает максимально допустимое значение.
-3	PACKET_LOSS	Обнаружена потеря пакета данных при передаче.

Коды ошибок сервера (коды error_num)

Код (hex)	Символьное имя	Описание
0x000	SUCCESS	Операция выполнена успешно.
0x001	NO_FOUND_TABLE	Запрашиваемая Modbus-таблица не найдена.
0x002	NOT_DESC_IN_ADDR_TABLE	Адрес не описан в таблице трансляции адресов.
0x003	ADDR_TRANSLATION_ARRAY_OUT	Запрошенное количество сигналов выходит за границы массива трансляции адресов (при чтении вектора).
0x004	NOT_READ_FROM_DB_NW	Не удалось выполнить чтение из базы данных NW.
0x005	NOT_WRITE_TO_DB_NW	Не удалось выполнить запись в базу данных NW.

8.

```
int Read_BitsData_From_DB_NW (int fd_sock, uint8_t cmd, int indx0, int nval,
uint8_t *bit_signals)
```

Используем данную функцию для чтения значений дискретных сигналов из базы данных (памяти) NW. Индекс `indx0` соответствует номеру сигнала в таблице с дискретными данными, `nval` количество сигналов в дискретной таблицы начиная с `indx0`, значения которых мы хотим получить из базы данных NW, `bit_signals` массив значений дискретных сигналов размерностью `nval` куда будут транслированы значения дискретных сигналов, запрошенных от удаленного сервера доступа к БД NW. Существует ограничения на количество запрашиваемых сигналов `nval` не должно превышать 11168 сигналов, это требование сформулировано исходя из необходимости разместить данные в пакете UDP размером 1400 Байт. В таблице 8 представлен запрос клиента серверу доступа к БД NW, а в таблице 9 ответ сервера.

Таблица 8 – Запрос клиента на чтение сигналов

Описание	Размер	Значения
Команда серверу на чтение сигналов из БД NW <code>cmd</code>	1 Байт	0x07 или произвольное значение
Начальный индекс сигнала в таблице <code>indx0</code>	<code>uint32_t</code>	Целое число
Количество считанных сигналов <code>nval</code>	<code>uint32_t</code>	Целое число
Случайное число <code>rand_mean = (rand() % 100);</code>	<code>uint8_t</code>	Целое число от 0 до 99

Таблица 9 – Ответ сервера, посылает заданные сигналы

Описание	Размер	Значения
Двух байтовое число 0xFFFF	<code>uint16_t</code> Байт	0xFFFF
Случайное число <code>rand_mean</code> полученное сервером при запросе от клиента см. таб.8	<code>uint8_t</code>	Целое число

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

Код ошибки сервера Error_num	uint8_t	Error_num код ошибки сервера, 0 запрос выполнен
Битовый поток значений дискретных сигналов размером nval	Битовый поток размерностью nval	Массив байт

В случае успешного выполнения функция возвращает 0.

Коды ошибок функции:

- 1 ошибка сети, тайм-аут;
- 2 размер запрошенных сигналов превышает допустимое значение;
- 3 потеря пакета;

Error_num код ошибки сервера;

9.

```
int Write_BitsData_To_DB_NW (int fd_sock, uint8_t cmd, int indx0, int nval,
uint8_t *bit_signals)
```

Используем данную функцию для обновления значений дискретных сигналов в базе данных (памяти) NW. Индекс indx0 соответствует номеру сигнала в таблице с дискретными данными, nval количество сигналов в дискретной таблицы начиная с indx0, значения которых мы хотим обновить в базе данных NW, bit_signals массив значений дискретных сигналов размерностью nval которые будут транслированы серверу доступа к БД NW для перезаписи новых значений сигналов в БД NW. Существует ограничения на количество обновленных сигналов nval не должно превышать 11120 сигналов, это требование сформулировано исходя из необходимости разместить данные в пакете UDP размером 1400 Байт. В таблице 10 представлен запрос клиента серверу доступа к БД NW, а в таблице 11 ответ сервера.

Таблица 10 – Запрос клиента на чтение сигналов

Описание	Размер	Значения
Команда серверу на чтение сигналов из БД NW cmd	1 Байт	0x0A или произвольное значение
Начальный индекс сигнала в таблице indx0	uint32_t	Целое число

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

Количество считанных сигналов nval	uint32_t	Целое число
Случайное число rand_mean = (rand () % 100);	uint8_t	Целое число от 0 до 99
Битовый поток значений дискретных сигналов размером nval	Битовый поток размерностью nval	Массив байт

Таблица 11 – Ответ сервера, посылает заданные сигналы

Описание	Размер	Значения
Двух байтовое число 0xFFFF	2 Байт	0xFFFF
Случайное число rand_mean полученное сервером при запросе от клиента см. таб.10	uint8_t	Целое число
Код ошибки сервера Error_num	uint8_t	Error_num код ошибки сервера, 0 запрос выполнен

В случае успешного выполнения функция возвращает 0.

Коды ошибок функции:

-1 ошибка сети, тайм-аут;

-2 размер запрошенных сигналов превышает допустимое значение;

-3 потеря пакета;

Error_num код ошибки сервера;

10.

```
int Read_IntData_From_DB_NW (int fd_sock, uint8_t cmd, int indx0, int nval, void
*int_signals)
```

Используем данную функцию для чтения значений целых сигналов из базы данных (памяти) NW. Индекс indx0 соответствует номеру сигнала в таблице с целыми данными, nval количество сигналов в дискретной таблицы с indx0, значения которых мы хотим получить из базы данных NW, int_signals массив значений целых сигналов размерностью nval куда будут транслированы значения дискретных сигналов, запрошенных от удаленного сервера доступа к БД NW. Если значение cmd будет принимать значение 0x3 или 0x4, в этом случае массив int_signals должен иметь тип int16_t. Существует ограничения на количество запрашиваемых сигналов nval не должно превышать 349 сигналов, это требование сформулировано исходя из необходимости разместить данные в пакете UDP размером 1400 Байт. В таблице 12 представлен запрос клиента серверу доступа к БД NW, а в таблице 13 ответ сервера.

Таблица 12 – Запрос клиента на чтение сигналов

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
---	---	-----

Описание	Размер	Значения
Команда серверу на чтение сигналов из БД NW cmd	1 Байт	0x08, или произвольное значение
Начальный индекс в таблице indx0	uint32_t	Целое число
Количество считанных сигналов nval	uint32_t	Целое число
Случайное число rand_mean = (rand() % 100);	uint8_t	Целое число от 0 до 99

Таблица 13 – Ответ сервера

Описание	Размер	Значения
Двух байтовое число 0xFFFF	2 Байт	0xFFFF
Случайное число rand_mean полученное сервером при запросе от клиента см. таб.12	uint8_t	Целое число
Код ошибки сервера Error_num	uint8_t	Error_num код ошибки сервера, 0 запрос выполнен
Массив значений целых сигналов размером nval	Массив int32_t размерностью nval	Массив int32_t [nval]

В случае успешного выполнения функция возвращает 0.

Коды ошибок функции:

- 1 ошибка сети, тайм-аут;
- 2 размер запрошенных сигналов превышает допустимое значение;
- 3 потеря пакета;

Error_num код ошибки сервера;

11.

```
int Write_IntData_To_DB_NW (int fd_sock, uint8_t cmd, int indx0, int nval, void *bit_signals)
```

Используем данную функцию для обновления значений целых сигналов в базе данных (памяти) NW. Индекс indx0 соответствует номеру сигнала в таблице с целыми данными, nval количество сигналов в дискретной таблицы с indx0, значения которых мы хотим обновить в базе данных NW, int_signals массив значений целых сигналов размерностью nval которые будут транслированы серверу доступа к БД NW для перезаписи новых значений сигналов в БД NW. Если значение cmd будет принимать значение 0x6 или

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
---	---	-----

0x10, в этом случае массив `int_signals` должен иметь тип `int16_t`. Существует ограничение на количество обновленных сигналов `nval` не должно превышать 347 сигналов, это требование сформулировано исходя из необходимости разместить данные в пакете UDP размером 1400 Байт. В таблице 14 представлен запрос клиента серверу доступа к БД NW, а в таблице 15 ответ сервера.

Таблица 14 – Запрос клиента на чтение сигналов

Описание	Размер	Значения
Команда серверу на чтение сигналов из БД NW <code>cmd</code>	1 Байт	0x0B, или произвольное значение
Начальный индекс в таблице <code>indx0</code>	<code>uint32_t</code>	Целое число
Количество считанных сигналов <code>nval</code>	<code>uint32_t</code>	Целое число
Случайное число <code>rand_mean = (rand() % 100);</code>	<code>uint8_t</code>	Целое число от 0 до 99
Массив значений целых сигналов размером <code>nval</code>	Массив <code>int32_t</code> размерностью <code>nval</code>	Массив <code>int32_t</code> [<code>nval</code>]

Таблица 15 – Ответ сервера

Описание	Размер	Значения
Двух байтовое число 0xFFFF	2 Байт	0xFFFF
Случайное число <code>rand_mean</code> полученное сервером при запросе от клиента см. таб.14	<code>uint8_t</code>	Целое число
Код ошибки сервера <code>Error_num</code>	<code>uint8_t</code>	Целое число

В случае успешного выполнения функция возвращает 0
 Коды ошибок функции:

- 1 ошибка сети, тайм-аут;
- 2 размер запрошенных сигналов превышает допустимое значение;
- 3 потеря пакета;

`Error_num` код ошибки сервера;

12.

```
int Read_DoubleData_From_DB_NW (int fd_sock, int indx0, int nval, double
*double_signals)
```

Используем данную функцию для чтения значений вещественных сигналов из базы данных (памяти) NW. Индекс `indx0` соответствует номеру сигнала в файле «discrets.dat»,

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
---	---	-----

nval количество сигналов в файле «analog.dat» начиная с indx0, значения которых мы хотим получить из базы данных NW, double_signals массив значений вещественных сигналов размерностью nval куда будут транслированы значения вещественных сигналов, запрошенных от удаленного сервера доступа к БД NW. Существует ограничения на количество запрашиваемых сигналов nval не должно превышать 174 сигналов, это требование сформулировано исходя из необходимости разместить данные в пакете UDP размером 1400 Байт. В таблице 16 представлен запрос клиента серверу доступа к БД NW, а в таблице 17 ответ сервера.

Таблица 16 – Запрос клиента на чтение сигналов

Описание	Размер	Значения
Команда серверу на чтение сигналов из БД NW cmd	1 Байт	0x09
Начальный индекс в таблице indx0	uint32_t	Целое число
Количество считанных сигналов nval	uint32_t	Целое число
Случайное число rand_mean = (rand() % 100);	uint8_t	Целое число от 0 до 99

Таблица 17 – Ответ сервера, посылает заданные сигналы

Описание	Размер	Значения
Двух байтовое число 0xFFFF	2 Байт	0xFFFF
Случайное число rand_mean полученное сервером при запросе от клиента см. таб.16	uint8_t	Целое число
Массив значений вещественных сигналов размером nval	Массив double размерностью nval	Массив double [nval]

В случае ошибки функцией будет возвращено значение меньше нуля, в случае успешного выполнения положительное значение.

В случае успешного выполнения функция возвращает 0

Коды ошибок функции:

- 1 ошибка сети, тайм-аут;
- 2 размер запрошенных сигналов превышает допустимое значение;
- 3 потеря пакета;

Error_num код ошибки сервера;

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

13.

```
int Write_DoubleData_To_DB_NW (int fd_sock, int indx0, int nval, double
*double_signals)
```

Используем данную функцию для обновления значений вещественных сигналов в базе данных (памяти) NW. Индекс `indx0` соответствует номеру сигнала в файле «discrets.dat», `nval` количество сигналов в файле «int.dat» начиная с `indx0`, значения которых мы хотим обновить в базе данных NW, `int_signals` массив значений вещественных сигналов размерностью `nval` которые будут транслированы серверу доступа к БД NW для перезаписи новых значений сигналов в БД NW. Существует ограничения на количество обновленных сигналов `nval` не должно превышать 173 сигналов, это требование сформулировано исходя из необходимости разместить данные в пакете UDP размером 1400 Байт. В таблице 18 представлен запрос клиента серверу доступа к БД NW, а в таблице 19 ответ сервера.

Таблица 18 – Запрос клиента на чтение сигналов

Описание	Размер	Значения
Команда серверу на чтение сигналов из БД NW <code>cmd</code>	1 Байт	0x0C
Начальный индекс сигнала в файле списка имен сигналов <code>indx0</code>	<code>uint32_t</code>	Целое число
Количество считанных сигналов <code>nval</code>	<code>uint32_t</code>	Целое число
Случайное число <code>rand_mean = (rand() % 100);</code>	<code>uint8_t</code>	Целое число от 0 до 99
Массив значений вещественных сигналов размером <code>nval</code>	Массив <code>double</code> размерностью <code>nval</code>	Массив <code>doublet [nval]</code>

Таблица 19 – Ответ сервера

Описание	Размер	Значения
Двух байтовое число 0xFFFF	2 Байт	0xFFFF
Случайное число <code>rand_mean</code> полученное сервером при запросе от клиента см. таб.18	<code>uint8_t</code>	Целое число
Код возврата	<code>int32_t</code>	Целое число

 3VS SOFTWARE DEVELOPMENT	Спецификация открытого протокола обмена NordWind Open UDP (Версия 2.6)	B01
--	---	-----

В случае ошибки функцией будет возвращено значение меньше нуля, в случае успешного выполнения положительное значение.

В случае успешного выполнения функция возвращает 0.

Коды ошибок функции:

-1 ошибка сети, тайм-аут;

-2 размер запрошенных сигналов превышает допустимое значение;

-3 потеря пакета;

Error_num код ошибки сервера;

4 АДАПТАЦИЯ БИБЛИОТЕКИ ПРОТОКОЛА К ПРОПУСКНЫМ ВОЗМОЖНОСТЯМ КАНАЛА СВЯЗИ

В случае низкой пропускной способности сети для уменьшения вероятности потери данных существует возможность изменять скорость отправки пакетов по протоколу UDP, для этого необходимо изменить время «сна» после отправки пакета в сеть, значение времени сна (в микросекундах) присвоено константе «TIME_DELTA» в файле «nw_exchange.h».