

Регистратор – 0.3

2012 г.

Содержание

1 Общие сведения	4
1.1 Назначение	4
1.2 Основные функции системы	4
1.3 Основные элементы системы	4
2 Руководство пользователя	5
2.1 Программа «Editor»	5
2.1.1 Основные сведения	5
2.1.2 Дерево элементов конфигурации	5
2.1.3 Панель элементов управления	6
2.1.4 Панель настройки com-порта	6
2.1.5 Панель настройки опроса прибора	8
2.1.6 Панель подключения модулей устройств	9
2.1.7 Панель подключения модулей экспорта данных	10
2.1.8 Поиск позиции	11
2.2 Программа «ReaderGUI»	11
2.2.1 Основные сведения	11
2.2.2 Вкладка значений результата опроса	12
2.2.3 Вкладка состояния опроса	13
2.2.4 Вкладка состояния экспорта	14
2.2.5 Параметры ini-файла	15
2.3 Программа «ReaderSvc»	15
2.3.1 Основные сведения	15
2.3.2 Управление состоянием службы «ReaderSvc»	15
2.3.2.1 Программа «Службы»	15
2.3.2.2 Программа «sc»	16
2.4 Программа «ReaderSvcCtrl»	17
2.5 Программа «TestExport»	17
2.5.1 Основные сведения	17
2.5.2 Панель элементов управления	18
2.5.3 Панель значений	18
2.5.4 Дополнительная информация	19
2.6 Программа «TestRequest»	20
2.6.1 Основные сведения	20
2.6.2 Вкладка настройки опроса	21
2.6.3 Вкладка информации о модуле	22
2.6.4 Вкладка тестового опроса	23
2.6.5 Вкладка отображения отправляемых и получаемых пакетов	24
2.7 Конфигурирование и опрос устройств	24
2.7.1 Выбор модулей устройств	24
2.7.2 Конфигурирование	25
2.7.3 Опрос	26
3 Руководство разработчика	27
3.1 Модуль (плагин) устройства	27
3.1.1 Функция GetInfo	27
3.1.2 Функция GetPackage	28
3.1.3 Функция GetData	28
3.2 Модуль (плагин) экспорта	29

3.2.1	Функция About	29
3.2.2	Функции Begin и End	30
3.2.3	Функции Export	30
3.3	Дополнительная информация	31
3.3.1	Номера ошибок	31
3.3.2	Пример модуля устройства	31
3.3.3	Пример модуля экспорта	35
3.3.4	Инструментальные средства использованные в проекте	37

1 Общие сведения

1.1 Назначение

Программный комплекс «Регистратор» предназначен для сбора оперативных данных (текущих значений) с различных видов регистрирующих устройств, подключенных к последовательному порту. Возможность опроса того или иного оборудования, как и выгрузка данных, осуществляется по принципу модульной (плагиной) архитектуры и расширяется за счёт них. Кроме основного программного обеспечения, ориентированного на сбор данных, в комплекс входит также программы настройки и тестирования.

1.2 Основные функции системы

К основным функциям системы относятся:

- опрос регистрирующих устройств через последовательный порт (com-порт);
- выгрузка полученных оперативных данных.

1.3 Основные элементы системы

К основным элементам системы относятся:

1. Editor – редактирование конфигураций опроса (см. описание 2.1);
2. ReaderGUI – опрос устройств в графическом режиме (см. описание 2.2);
3. ReaderSvc – опрос устройств в фоновом режиме. Регистрируется в Windows как служба (см. описание 2.3);
4. ReaderSvcCtrl – управление сервисом ReaderSvc (см. описание 2.4);
5. TestExport – тестирование выгрузки данных выбранным модулем (см. описание 2.5);
6. TestRequest – тестирование опроса прибора выбранным модулем (см. описание 2.6).

2 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

2.1 Программа «Editor»

2.1.1 Основные сведения

Программа «Editor» служит для редактирования конфигураций опроса. Для её запуска необходимо выбрать: *Пуск* → *Программы* → *Registrar* → *Editor*. Общий вид программы представлен на рисунке 1.

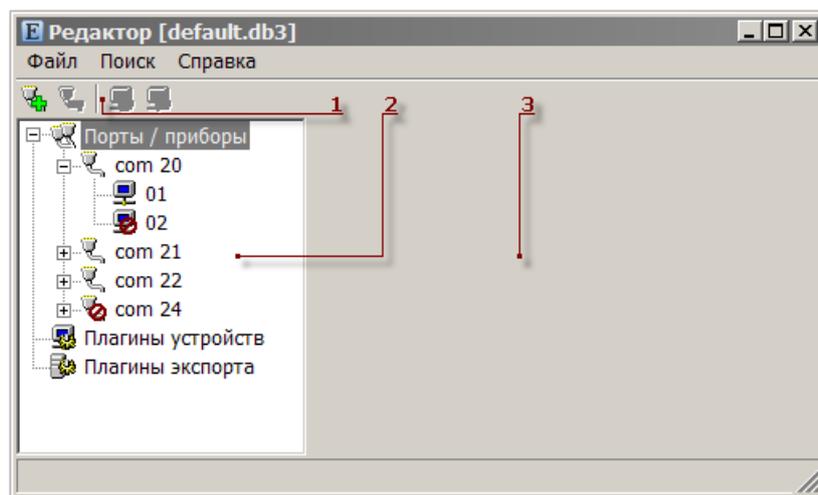


Рисунок 1 – Общий вид программы «Editor»

Программа состоит из следующих частей:

- панель элементов управления (рисунок 1-1);
- дерево элементов конфигурации (рисунок 1-2);
- свойства текущего элемента конфигурации (рисунок 1-3);

2.1.2 Дерево элементов конфигурации

Дерево элементов конфигурации представлено на рисунке 2. Описание каждого из элементов дерева описаны в таблице 1.

Таблица 1 – Элементы дерева конфигурации

Элемент	Назначение
 Порты / приборы	Элемент содержит в себе элементы com-портов и устройств. Служит для объединения групп.
 com №	Настройка com-порта указанного номера. Содержит в себе элементы устройств.
 №	Настройка параметров опроса прибора. Включает в себя настройку каналов опроса.
 Плагины устройств	Управление модулями устройств.
 Плагины экспорта	Управление модулями выгрузки данных.

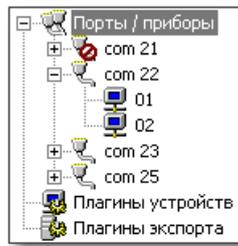


Рисунок 2 – Дерево элементов конфигурации

2.1.3 Панель элементов управления

Панель элементов управления находится в верхней части главного окна программы и содержит кнопки, необходимые для выполнения действий, соответствующих текущему элементу в дереве конфигурации (рисунок 3).



Рисунок 3 – Панель элементов управления

Таблица 2 – Элементы панели управления

Элемент	Назначение
	Добавление в опрос новый com-порт.
	Удаление com-порта вместе со всеми связными приборами.
	Добавление в опрос новый прибор.
	Удаление прибора.

2.1.4 Панель настройки com-порта

Панель настройки com-порта (см. рисунок 4) позволяет задать настройки для подключения. Данная панель появится при выборе элемента дерева конфигурации «com №» (см. рисунок 2). Описание основных компонент панели описаны в таблице 3.

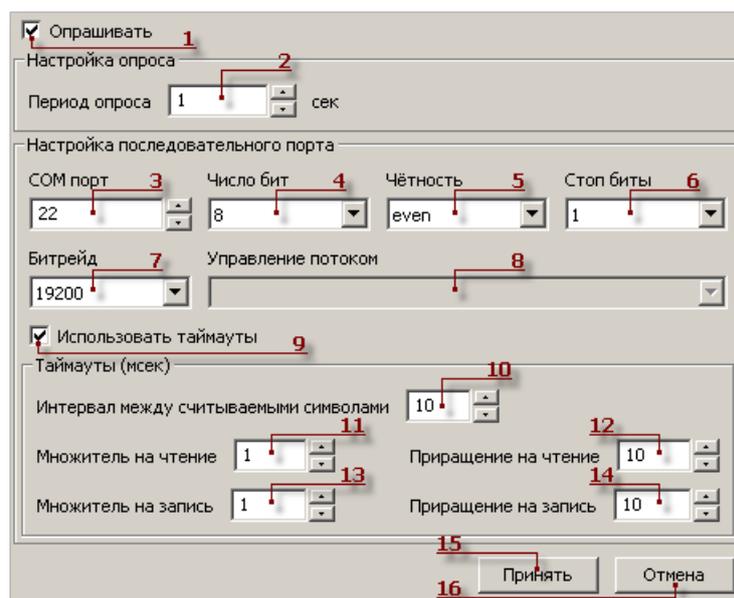


Рисунок 4 – Панель настройки com-порта

Таблица 3 – Описание компонент панели настройки com-порта

№ на рис. 4	Назначение
1	Включение/отключение опроса текущего com-порта. Если опрос отключён, то соответствующий порт, вместе со всеми связными устройствами, опрашиваться не будет.
2	Интервал между началами опроса. Если время опроса превышает данный интервал (период), то следующий опрос начнётся незамедлительно.
3	Номер com-порта.
4	Определяет число информационных бит в передаваемых и принимаемых байтах. Число информационных бит может быть в диапазоне от 5 до 8.
5	Режимы контрольного бита. Может принимать значения: <ul style="list-style-type: none"> 1. none – проверка на паритет не используется и бит не выставляется; 2. even – проверка на чётность, дополняет передаваемый символ так, чтобы количество единиц в передаваемом символе было четным; 3. odd – проверка на нечётность, дополняет передаваемый символ так, чтобы количество единиц в передаваемом символе было нечетным; 4. mark – бит паритета всегда равен единице
6	Задаёт количество стоповых бит. Может принимать значения: 1, 1.5 или 2 стоповых бита.
7	Скорость передачи данных. Возможно задание следующих значений: 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200, 128000, 256000 бод.
8	Управление потоком ¹ .
9	Включение/отключение использования таймаутов. Если данная опция отключена, то программа будет ждать приём всех байт заявленных в модуле и таким образом может остановить опрос по данному com-порту.
10	Значение, задающее максимальное время (в миллисекундах) интервала между поступлением двух символов в коммуникационную линию. Если интервал между поступлением любых двух символов будет больше этой величины, операция чтения завершается и любые буферизированные данные возвращаются.
11, 13	Множитель, используемый, чтобы вычислить полный период времени простоя для операций чтения/записи, в миллисекундах. Для каждой операции чтения/записи, это значение умножается на затребованное число байтов, которые читаются/передаются.

¹На данном этапе развития функция не реализована

12, 14	Константа, используемая, чтобы вычислить полный (максимальный) период времени простоя для операций чтения/записи, в миллисекундах. Для каждой операции чтения/записи, это значение добавляется к произведению множителя и прочитанного/записанного числа байтов.
15, 16	Принять/отменить внесённые изменения. Становится активной при изменении какого-либо параметра.

2.1.5 Панель настройки опроса прибора

Панель настройки конфигурации прибора изображена на рисунке 5. Кроме типа прибора и его наименования, здесь также задаются все опрашиваемые каналы. В таблице 4 представлено описание всех компонент данной панели настроек.

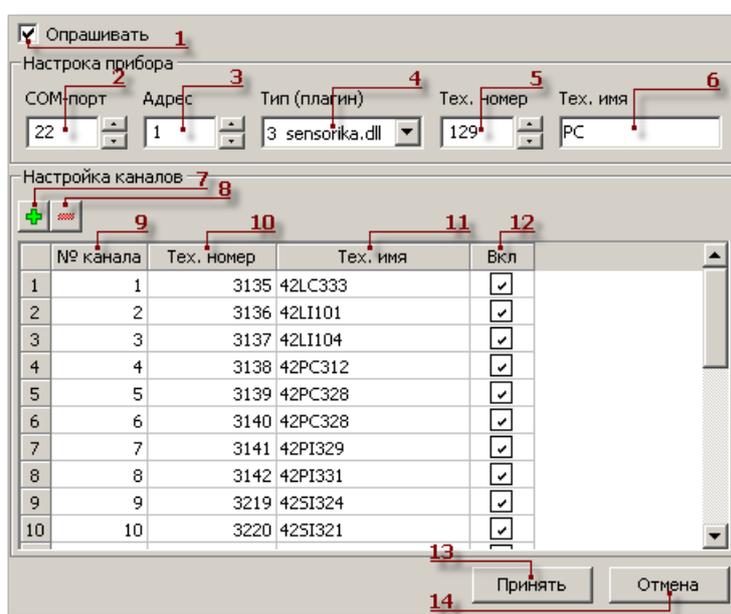


Рисунок 5 – Панель настройки опроса прибора

Таблица 4 – Описание компонент панели настройки опроса прибора

№ на рис. 5	Назначение
1	Включение/отключение опроса текущего прибора.
2	Номер com-порта на котором располагается данный прибор.
3	Адрес опрашиваемого прибора.
4	Тип прибора. Точнее модуль, с помощью которого будет выполняться формирование и обработка данных. В списке появляются только те модули, которые добавлены в панели подключения модулей устройств (см. описание 2.1.6).
5	Технологический номер прибора. Не обязательный параметр и может быть опущен.
6	Технологическое имя прибора. Не обязательный параметр и может быть опущен.
7	Добавление нового канала.

8	Удаление канала. Кнопка будет активна только тогда, когда будет выбран канал при помощи щелчка мыши по крайнему левому нумерованному полю.
9	Поле номера канала. Является обязательным для заполнения.
10	Технологический номер позиции. Является не обязательным.
11	Технологическое имя позиции. Является не обязательным.
12	Включение/отключение опроса текущего канала.
13, 14	Принять/отменить внесённые изменения. Становится активной при изменении какого-либо параметра.

2.1.6 Панель подключения модулей устройств

Данная панель предоставляет возможность подключить необходимые модули устройств и просмотреть их характеристики (рисунок 6). В таблице 5 представлено описание всех компонент данной панели настроек.

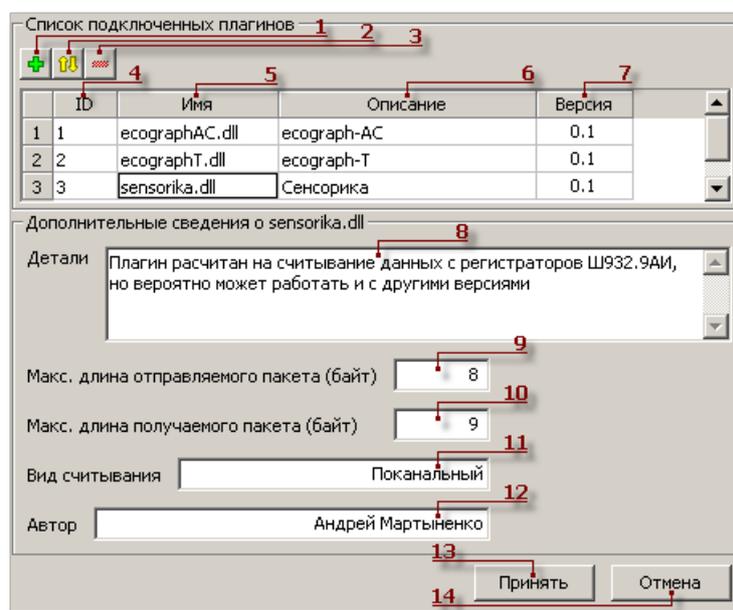


Рисунок 6 – Панель подключения модулей устройств

Таблица 5 – Описание компонент панели подключения модулей устройств

№ на рис. 6	Назначение
1	Добавление нового модуля.
2	Замена модуля. При замене, все устройства, опрашиваемые заменяемым модулем, будут опрашиваться заменяющим.
3	Удаление модуля. Удаление будет возможно, только если не будет устройств, зависящих от данного модуля. Кнопка будет активна только тогда, когда будет выбран модуль (или группа модулей) при помощи щелчка мыши по крайнему левому нумерованному полю.
4	Идентификационный номер модуля ² .
5	Имя модуля в файловой системе.

²Показывается пользователю для того, чтобы не было путаницы при замене модулей

6	Название модуля данное разработчиком.
7	Версия модуля.
8	Детальное описание модуля. Автор может описать всё то, что он считает нужным.
9,10	Максимальная длина отправляемого/получаемого пакета ³ .
11	Информирует, как будет осуществляться опрос устройства. Опрашивать может как поканально, так и все каналы за один запрос.
12	Автор модуля.
13, 14	Принять/отменить внесённые изменения. Становится активной при изменении какого-либо параметра.

Если при загрузке модуля происходит ошибка, то строка с описанием данного модуля подсвечивается красным цветом. Соответственно, при переходе на данную строку, никакой дополнительной информации выводиться не будет.

2.1.7 Панель подключения модулей экспорта данных

Панель экспорта данных предоставляет возможность подключить необходимые модули экспорта и просмотреть их характеристики (рисунок 7). В таблице 6 представлено описание всех компонент данной панели настроек.

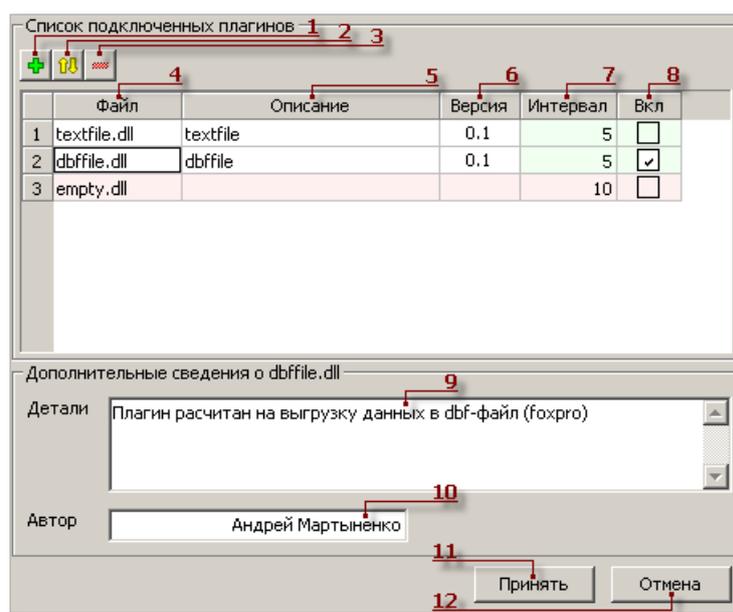


Рисунок 7 – Панель подключения модулей экспорта

Таблица 6 – Описание компонент панели подключения модулей устройств

№ на рис. 7	Назначение
1	Добавление нового модуля.
2	Замена модуля.
3	Удаление модуля. Кнопка будет активна только тогда, когда будет выбран модуль (или группа модулей) при помощи щелчка мыши по крайнему левому нумерованному полю.

³ Данная информация необходима в большей мере системе, нежели пользователю

4	Имя модуля в файловой системе.
5	Название модуля данное разработчиком.
6	Версия модуля.
7	Интервал, через который будет выполняться выгрузка (экспорт) данных. Интервал задаётся в секундах.
8	Включение/отключение выполнения выгрузки данных.
9	Детальное описание модуля. Автор может описать всё то, что он считает нужным.
10	Автор модуля.
11, 12	Принять/отменить внесённые изменения. Становится активной при изменении какого-либо параметра.

Поля, подсвеченные зелёным цветом, могут редактироваться (колонки под номерами 7 и 8 на рисунке 7). Если при загрузке модуля происходит ошибка, то строка с описанием данного модуля подсвечивается красным цветом. Соответственно, при переходе на данную строку, никакой дополнительной информации выводиться не будет.

2.1.8 Поиск позиции

Для поиска позиции необходимо выбрать в меню «Поиск → Поиск канала», либо просто нажать сочетание клавиш *ctrl+f*. При этом появится окно поиска изображённое на рисунке 8. Как видно из данного рисунка, можно выбрать режим поиска: либо поиск по технологическому номеру, либо по технологическому имени. Если поиск будет успешным, то программа переключится на панель настройки опроса прибора и установит курсор в таблице каналов на найденную позицию. Если поиск не увенчается успехом, то программа сообщит об этом⁴. Так как технологический номер и технологическое имя являются уникальными то понятия далее, в режиме поиска, отсутствуют.

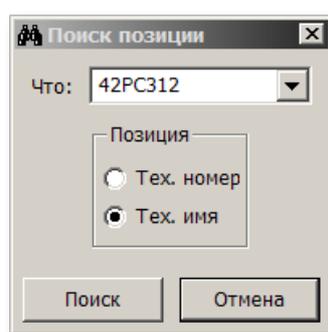


Рисунок 8 – Окно поиска позиции

2.2 Программа «ReaderGUI»

2.2.1 Основные сведения

Программа «ReaderGUI» позволяет осуществлять опрос приборов и выполнять экспорт данных. Для её запуска необходимо выбрать: *Пуск → Программы → Registrator → ReaderGUI*. Общий вид программы представлен на рисунке 9.

⁴На данном этапе развития проекта, поиск является строгим и регистрозависимым

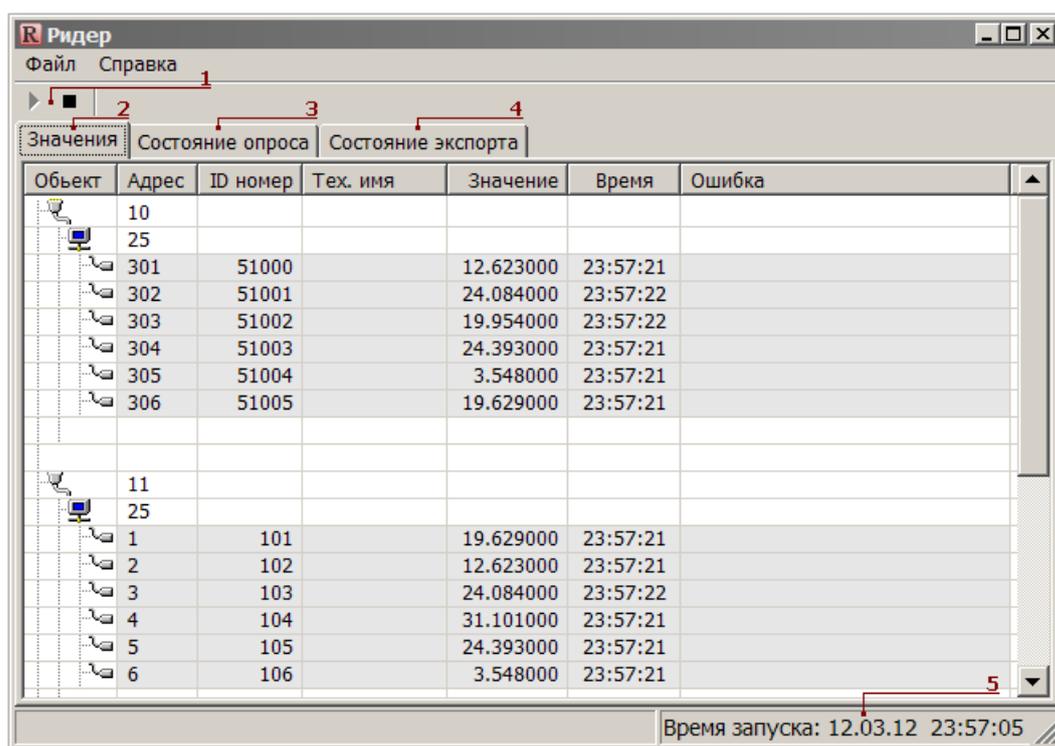


Рисунок 9 – Общий вид программы «ReaderGUI»

Программа состоит из следующих частей:

- панель элементов управления запуска/останова опроса (рисунок 9-1);
- вкладка результата опроса (рисунок 9-2);
- вкладка состояния опроса (рисунок 9-3);
- вкладка состояния экспорта (рисунок 9-4);
- панель отображения времени запуска/останова опроса (рисунок 9-5).

2.2.2 Вкладка значений результата опроса

Вкладка значений изображена на рисунке 10. Здесь отображается результат опроса всех каналов. Описание всех компонент данной вкладки представлено в таблице 7.

Таблица 7 – Описание вкладки значений результата опроса

№ на рис. 10	Назначение
1	Тип объекта. Изображается в графическом виде.
2	Адрес объекта (адрес com-порта, прибора или канала).
3	Технологический номер канала (или просто ID). Данная информация выводится только напротив каналов.
4	Технологическое имя канала. Выводится только напротив каналов.
5	Значение, полученное при опросе данного канала.
6	Время, в которое было получено значение.
7	Ошибка, возникшая при опросе.

Объект	Адрес	ID номер	Тех. имя	Значение	Время	Ошибка
10						
25						
301		51000		12.623000	23:57:21	
302		51001		24.084000	23:57:22	
303		51002		19.954000	23:57:22	
304		51003		24.393000	23:57:21	
305		51004		3.548000	23:57:21	
306		51005		19.629000	23:57:21	

Рисунок 10 – Вкладка значений

2.2.3 Вкладка состояния опроса

Вкладка состояния опроса (см. рисунок 11) отображает текущий статус по каждому com-порту. Описание всех компонент данной вкладки представлено в таблице 7.

Сос...	Имя	Время запуска	Ожидание	Ошибка
▶	com-10	18.03.12 01:00:19	4	
▶	com-11	18.03.12 01:00:19	9	
■	com-12	18.03.12 01:00:18	1	Ошибка при открытии порта

Рисунок 11 – Вкладка состояния опроса

Таблица 8 – Описание вкладки состояния опроса

№ на рис. 11	Назначение
1	В графическом виде представлено текущее состояние. Состояния могут быть следующие: <ol style="list-style-type: none"> 1. ▶ – выполнение опроса; 2. ▶ – ожидание старта выполнения; 3. ■ – ошибка работы потока.
2	Номер (имя) com-порта, через который осуществляется опрос.
3	Время последней попытки опроса.
4	Ожидание в секундах, через которое произойдёт следующая попытка опроса.
5	Описание произошедшей ошибки выполнения опроса.

Возможна ситуация, когда в колонке с ожиданием (рисунок 11-4) будут фигурировать отрицательные числовые значения. Это говорит о том, что программа опроса не укладывается в заданный для неё интервал времени. Для исправления такой ситуации, необходимо попробовать выполнить следующее:

- уменьшить количество позиций опрашиваемых через данный порт;
- увеличить скорость передачи данных;
- подкорректировать таймауты в сторону уменьшения, в том случае если они используются.

2.2.4 Вкладка состояния экспорта

Вкладка состояния опроса (см. рисунок 12) отображает текущий статус по каждому плагину. Описание всех компонент данной вкладки представлено в таблице 9.



Рисунок 12 – Вкладка состояния экспорта

Таблица 9 – Описание вкладки состояния экспорта

№ на рис. 12	Назначение
1	В графическом виде представлено текущее состояние. Состояния могут быть следующие: <ol style="list-style-type: none"> 1. ► – выполнение экспорта; 2. ► – ожидание старта эспорта; 3. ■ – ошибка работы потока.
2	Имя подключенного модуля (плагина).
3	Время последней попытки экспорта.
4	Ожидание в секундах, через которое произойдёт следующая попытка экспорта.
5	Описание произошедшей ошибки выполнения экспорта.

Как и в случае с опросом, в колонке с ожиданием (рисунок 12-4) могут фигурировать отрицательные числа. Это означает, что программа не успевает выгрузить значения за заданный промежуток времени.

2.2.5 Параметры ini-файла

После первого запуска программа ReaderGUI создаст файл *Reader.ini*. Данный файл может быть откорректирован только вручную. Пример данного файла описан в листинге 1.

Листинг 1 – Параметры Reader.ini

```
[general]
database=config/default.db3

[gui]
autostart=false
redraw_window=5
```

Именно с такими параметрами по умолчанию создаётся *Reader.ini*. Секция *general* содержит всего один параметр – *database*. Это параметр устанавливает путь к конфигурационной базе данных. Данный параметр необходим не только программе ReaderGUI, но и программе ReaderSvc (см. описание 2.3), которая устанавливается как служба Windows. Если этот параметр не задан, то будет произведена попытка подключить базу по умолчанию – *default.db3*. Секция *gui* используется только программой ReaderGUI. Параметр *autostart* задаёт автоматический запуск опроса. Может принимать значение *true* или *false*. Параметр *redraw_window* указывает время в секундах обновления данных в таблице.

2.3 Программа «ReaderSvc»

2.3.1 Основные сведения

Данная программа является сервисной и устанавливается в ОС Windows как служба. Полное название службы – ReaderSvc(Registrar)⁵. Программа может быть запущена со следующими параметрами (таблица 10):

Таблица 10 – Параметры запуска программы «ReaderSvc»

Параметр	Назначение
install	Зарегистрировать службу ReaderSvc.
uninstall	Удалить службу ReaderSvc.

2.3.2 Управление состоянием службы «ReaderSvc»

Управление запуском и остановом службы может осуществляться следующими способами:

- через программу «Службы» в ОС Windows (см. описание 2.3.2.1);
- через командную строку, с помощью команды *sc* (см. описание 2.3.2.2);
- через программу «ReaderSvcCtrl» (см. описание 2.4).

2.3.2.1 Программа «Службы»

Для запуска программы службы, необходимо выбрать: *Пуск* → *Система* → *Настройка* → *Администрирование* → *Службы*. Либо нажать сочетание клавиш *win+r* и в появившемся окне «Запуск программы» набрать *services.msc*, после чего нажать *enter*. В ре-

⁵В дальнейшем чаще будет фигурировать просто как ReaderSvc

зультате появится окно программы «Службы», где будет представлен список всех служб, включая установленную ReaderSvc (рисунок 13).

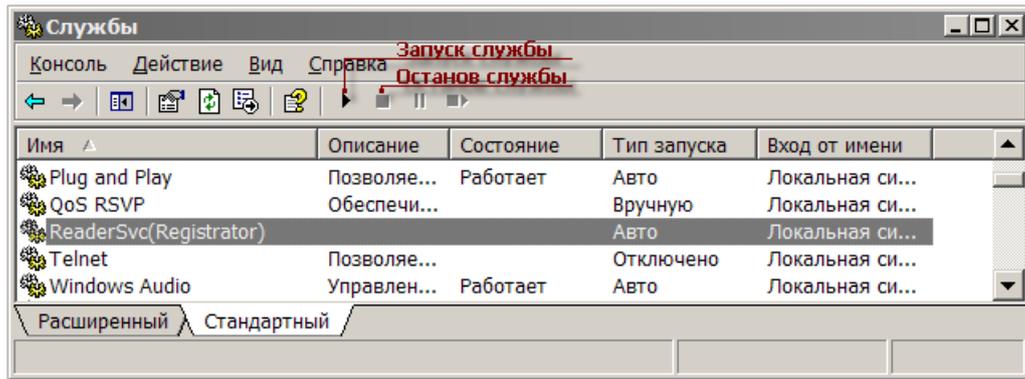


Рисунок 13 – Программа «Службы» в Windows

С помощью данной программы можно посмотреть текущее состояние службы, а также выполнить её запуск и останов.

2.3.2.2 Программа «sc»

Для того, чтобы воспользоваться данной программой необходимо запустить командную строку. Это можно сделать, к примеру, нажав комбинацию *win+r* и в появившемся окне «Запуск программы» набрать *cmd*, после чего нажать *enter*. Появится командная строка, в которой можно выполнить команду *sc* с нужными параметрами. Для запуска службы предназначен параметр *start*, а для останова – *stop*. Формат команды следующий: *sc <команда> <имя службы>*. В данном случае полное имя службы – *ReaderSvc(Registrator)*. На рисунке 14 показан результат выполнения обеих команд.

```
console
C:\>sc start ReaderSvc(Registrator)
SERVICE_NAME: ReaderSvc(Registrator)
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE                : 2  START_PENDING
                        (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE      : 0  (0x0)
        SERVICE_EXIT_CODE   : 0  (0x0)
        CHECKPOINT          : 0x0
        WAIT_HINT           : 0x7d0
        PID                 : 264
        FLAGS                :
C:\>sc stop ReaderSvc(Registrator)
SERVICE_NAME: ReaderSvc(Registrator)
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE                : 3  STOP_PENDING
                        (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE      : 0  (0x0)
        SERVICE_EXIT_CODE   : 0  (0x0)
        CHECKPOINT          : 0x0
        WAIT_HINT           : 0x0
C:\>
```

Рисунок 14 – Результат выполнения запуска и останова службы «ReaderSvc»

2.4 Программа «ReaderSvcCtrl»

Для запуска ПО «ReaderSvcCtrl» необходимо выбрать: *Пуск* → *Программы* → *Registrator* → *ReaderSvcCtrl*. Данная программа является консольной и служит для управления службой ReaderSvc. Внешний вид программы изображён на рисунке 15.

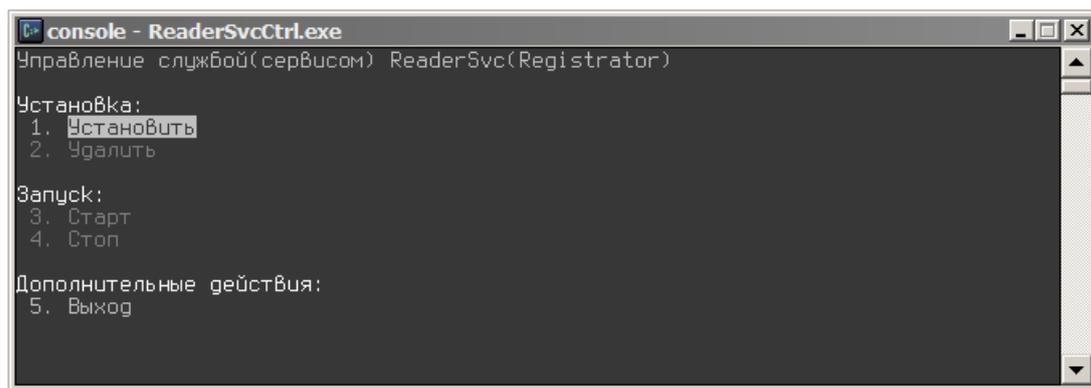


Рисунок 15 – Общий вид программы «ReaderSvcCtrl»

Программа может выполнить следующие действия над службой ReaderSvc(Registrator) ⁶:

1. Установить
2. Удалить
3. Запустить
4. Остановить

Для выхода из программы нужно выбрать пункт 5 или просто нажать клавишу *esc*.

Программа, при запуске, определяет состояние (статус) службы и предоставляет на выбор только те операции, которые могут быть выполнены. К примеру, если служба не установлена (как на рисунке 15), то пункты удаления, запуска и останова будут не активны.

2.5 Программа «TestExport»

2.5.1 Основные сведения

Программа «TestExport» служит для тестирования плагинов экспорта. Для её запуска необходимо выбрать: *Пуск* → *Программы* → *Registrator* → *TestExport*. Общий вид программы представлен на рисунке 16.

Программа состоит из следующих частей:

- панель элементов управления (рисунок 16-1);
- панель значений (рисунок 16-2);
- панель отображения имён загруженных файлов (рисунок 16-3).

⁶Описано по пунктам, так, как представлено на рисунке 15

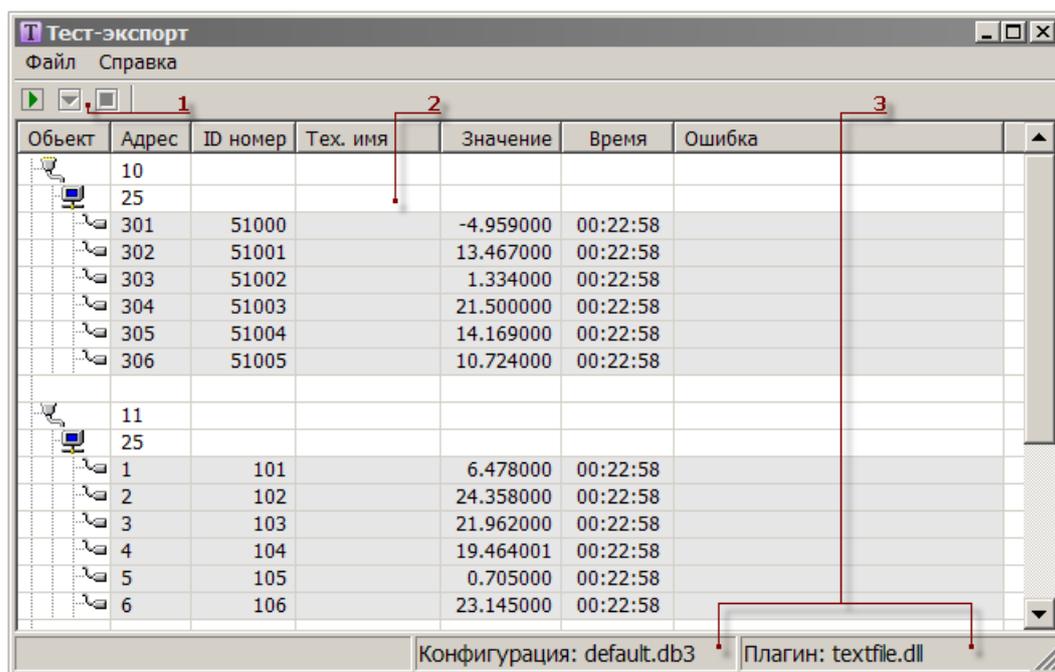


Рисунок 16 – Общий вид программы «TestExport»

2.5.2 Панель элементов управления

Панель элементов управления находится в верхней части главного окна программы и содержит кнопки, необходимые для выполнения функций описанных в плагине (рисунок 17). В таблице 11 представлено краткое описание каждой из кнопок.



Рисунок 17 – Панель элементов управления

Таблица 11 – Элементы панели управления

Элемент	Назначение
	Выполнить функцию “Begin”. Если функция не выполниться, то появится сообщение об ошибке. При этом остальные кнопки не станут активными.
	Выполнить функцию “Export”. Если функция вернёт ошибку, то появится соответствующее сообщение с её номером.
	Выполнить функцию “End”. Если произойдёт ошибка, то программа предложит либо проигнорировать ошибку, либо остаться в режиме экспорта.

2.5.3 Панель значений

Панель значений предназначена для задания и отображения экспортируемых данных (см. рисунок 18). Описание всех компонент данной панели представлено в таблице 12.

Таблица 12 – Описание панели значений

№ на рис. 18	Назначение
1	Тип объекта. Изображается в графическом виде.

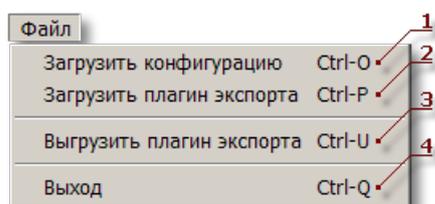
2	Адрес объекта (адрес com-порта, прибора или канала).
3	Технологический номер канала (или просто ID). Данная информация выводится только напротив каналов.
4	Технологическое имя канала. Выводится только напротив каналов.
5	Значение, которое будет экспортироваться. Для генерации случайных значений необходимо нажать левой клавишей мыши в заголовке колонки.
6	Время, которым будут датироваться все экспортируемые данные. Для задания текущего времени необходимо нажать левой клавишей мыши в заголовке колонки.
7	Генерация возможных ошибок. Для их появления необходимо нажать левой клавишей мыши в заголовке колонки. При повторном нажатии ошибки исчезнут.

Объект	Адрес	ID номер	Тех. имя	Значение	Время	Ошибка
10	25					
301		51000		-4.959000	00:22:58	
302		51001		13.467000	00:22:58	
303		51002		1.334000	00:22:58	
304		51003		21.500000	00:22:58	
305		51004		14.169000	00:22:58	
306		51005		10.724000	00:22:58	

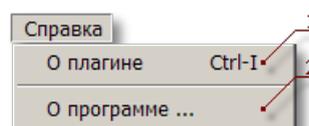
Рисунок 18 – Панель значений

2.5.4 Дополнительная информация

Для того, чтобы протестировать плагин экспорта необходимо загрузить имеющуюся конфигурацию опроса. Чтобы загрузить конфигурацию, необходимо, выбрать в меню «Файл → Загрузить конфигурацию» (см. рисунок 19.а-1) или просто нажать *ctrl+o*. Кроме этого нужно загрузить сам плагин экспорта, для этого необходимо выбрать в меню «Файл → Загрузить плагин экспорта» (см. рисунок 19.а-2) или нажать *ctrl+p*.



а) Пункт меню «Файл»



б) Пункт меню «Справка»

Рисунок 19 – Меню

После загрузки файлов, в панели статуса (см. рисунок 16-3), будут отображаться их имена. Как только конфигурация и плагин загружены, кнопка выполнения функции «Begin» на панели элементов управления (см. рисунок 17) становится активной.

Информацию о плагине можно узнать выбрав в меню «Справка → О плагине» (см. рисунок 19.б-1). После чего появиться окно в котором будет отображена вся информация (см. рисунок 20).

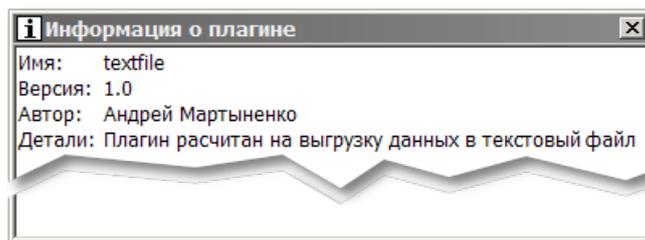


Рисунок 20 – Окно информации о плагине

2.6 Программа «TestRequest»

2.6.1 Основные сведения

Программа «TestRequest» служит для тестирования плагинов опроса устройств. Для её запуска необходимо выбрать: *Пуск → Программы → Registrator → TestRequest*. Общий вид программы представлен на рисунке 21.

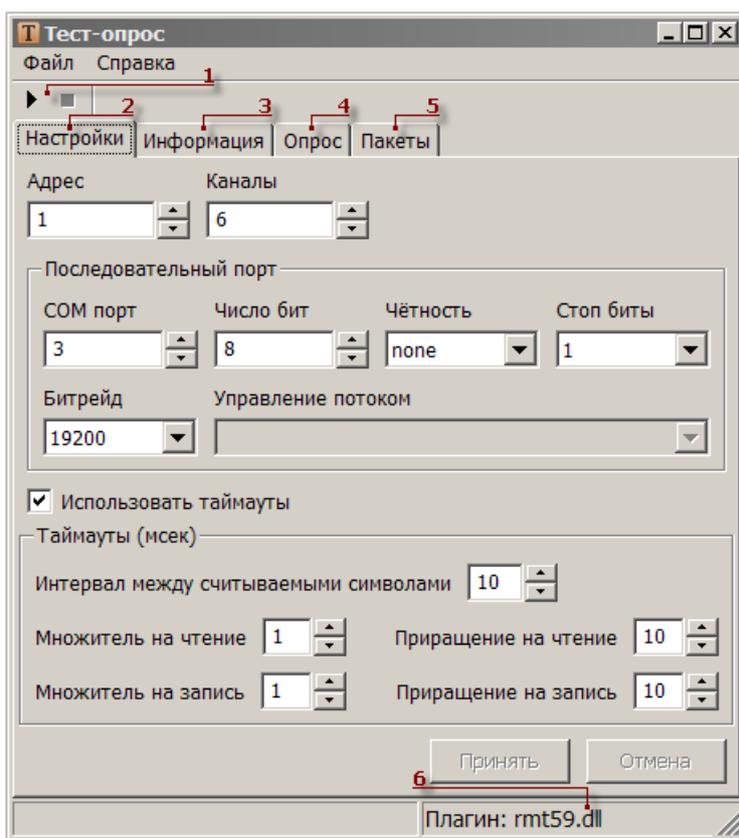


Рисунок 21 – Общий вид программы «TestRequest»

Программа состоит из следующих частей:

- вкладка настройки (рисунок 21-1);

- вкладка информации о выбранном модуле (плагине) (рисунок 21-2);
- вкладка тестового опроса (рисунок 21-3);
- вкладка отображения отправляемых и получаемых пакетов (рисунок 21-4).

2.6.2 Вкладка настройки опроса

Вкладка настроек изображена на рисунке 22. Описание всех компонент данной вкладки представлено в таблице 13.

Таблица 13 – Описание компонент вкладки настроек

№ на рис. 22	Назначение
1	Адрес опрашиваемого прибора.
2	Количество опрашиваемых каналов. Номера каналов не указываются. Каналы опрашиваются последовательно начиная с 1-го.
3	Номер сот-порта.
4	Определяет число информационных бит в передаваемых и принимаемых байтах. Число информационных бит может быть в диапазоне от 5 до 8.
5	Режимы контрольного бита. Может принимать значения: <ol style="list-style-type: none"> 1. none – проверка на паритет не используется и бит не выставляется; 2. even – проверка на четность, дополняет передаваемый символ так, чтобы количество единиц в передаваемом символе было четным; 3. odd - проверка на нечетность, дополняет передаваемый символ так, чтобы количество единиц в передаваемом символе было нечетным; 4. mark - бит паритета всегда равен единице.
6	Задаёт количество стоповых бит. Может принимать значения: 1, 1.5 или 2 стоповых бита.
7	Скорость передачи данных. Возможно указание следующих значений: 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200, 128000, 256000 бод.
8	Управление потоком ⁷ .
9	Включение/отключение использования таймаутов. Если данная опция отключена, то программа будет ждать приём всех байт заявленных в модуле и таким образом может остановить опрос по данному сот-порту.
10	Значение, задающее максимальное время (в миллисекундах) интервала между поступлением двух символов в коммуникационную линию. Если интервал между поступлением любых двух символов будет больше этой величины, операция чтения завершается и любые буферизированные данные возвращаются.

⁷На данном этапе развития проекта не используется

11, 13	Множитель, используемый, чтобы вычислить полный период времени простоя для операций чтения/записи, в миллисекундах. Для каждой операции чтения/записи, это значение умножается на затребованное число байтов, которые читаются/передаются.
12, 14	Константа, используемая, чтобы вычислить полный (максимальный) период времени простоя для операций чтения/записи, в миллисекундах. Для каждой операции чтения/записи, это значение добавляется к произведению множителя и прочитанного/записанного числа байтов.
15, 16	Принять/отменить внесённые изменения. Становится активной при изменении какого-либо параметра.

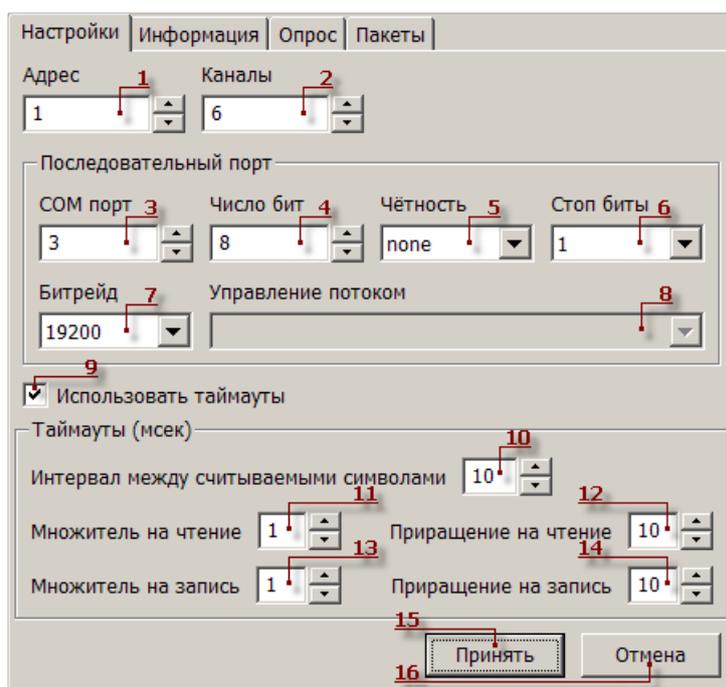


Рисунок 22 – Вкладка настройки опроса

2.6.3 Вкладка информации о модуле

Вкладка информации о модуле изображена на рисунке 23. Описание всех компонент данной вкладки представлено в таблице 14.

Таблица 14 – Описание компонент вкладки информации

№ на рис. 23	Назначение
1	Название модуля данное разработчиком.
2	Автор модуля.
3	Версия модуля.
4	Детальное описание модуля. Автор может описать всё то, что он считает нужным.
5, 6	Максимальная длина отправляемого/получаемого пакета. Данная информация необходима в большей мере системе, нежели пользователю.

7	Информирует, как будет осуществляться опрос устройства. Опрашивать может как поканально, так и путём опроса всех каналов за один запрос (группой).
---	----------------------------------------------------------------------------------------------------------------------------------------------------

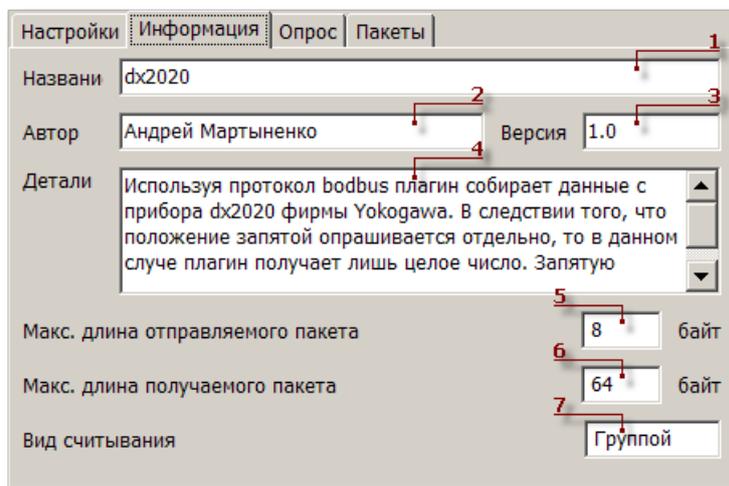


Рисунок 23 – Вкладка информации о модуле

2.6.4 Вкладка тестового опроса

Вкладка тестового опроса изображена на рисунке 24. Описание всех компонент данной вкладки представлено в таблице 15.

Номер	Значение	Ошибка	Время
01	0.000000	Таймаут	00:11:35
02	32.181999		00:11:35
03	23.856001		00:11:35
04	19.490000		00:11:35
05	0.000000	Ошибка проверки к...	00:11:35
06	4.324000		00:11:35
07	6.486000		00:11:35
08	19.677000		00:11:35
09	15.969000		00:11:35
10	11.643000		00:11:35

Рисунок 24 – Вкладка тестового опроса

Таблица 15 – Описание компонент вкладки тестового опроса

№ на рис. 24	Назначение
1	Номер опрашиваемого канала.
2	Значение, полученное от прибора по данному каналу.
3	Ошибка, полученная от прибора или появившаяся во время обработки данных.
4	Время, получения данных.

2.6.5 Вкладка отображения отправляемых и получаемых пакетов

Вкладка отображения отправляемых и получаемых пакетов изображена на рисунке 25. Описание всех компонент данной вкладки представлено в таблице 16.

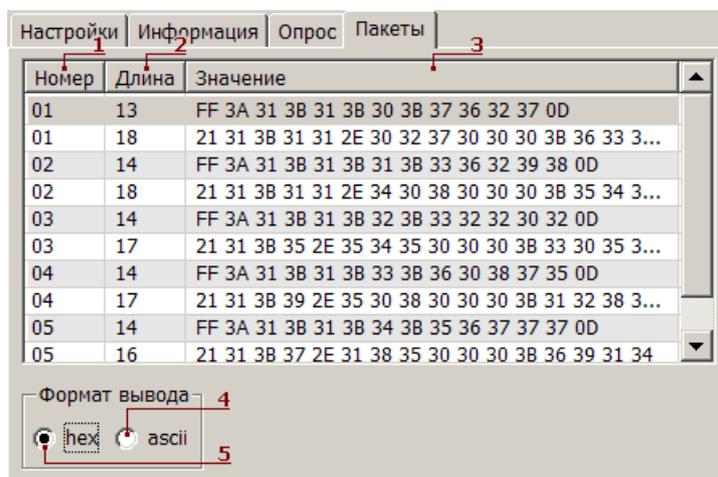


Рисунок 25 – Вкладка отображения отправляемых и получаемых пакетов

Таблица 16 – Описание компонент вкладки пакетов данных

№ на рис. 25	Назначение
1	Номер опрашиваемого канала.
2	Длина отправленного/полученного пакета.
3	Отправляемый/получаемый пакет.
4	Отображение пакетов в шестнадцатеричном виде.
5	Отображение пакетов в символьном виде.

Как видно из рисунка 25, на каждый канал приходится по два значения. Первое значение это отправляемый пакет, второе – получаемый пакет.

2.7 Конфигурирование и опрос устройств

2.7.1 Выбор модулей устройств

На первом этапе необходимо определиться с модулями устройств. Для того, чтобы проверить, подходит ли данный модуль к устройству, нужно запустить программу TestRequest (см. описание 2.6). Нажав в меню программы «Файл → Открыть» выбрать конкретный модуль. Как только выбор будет осуществлён, детальная информация по выбранному плагину будет отображена во вкладке «Информация» (см. описание 2.6.3). Если данный модуль подходит по описанию, то необходимо переключиться на вкладку «Настройка» (см. описание 2.6.2) и задать параметры подключения к com-порту и параметры опрашиваемого прибора. Как только настройка завершена, выполнить попытку опроса прибора. В случае если модуль устройства подходит к данному прибору, то в колонке «Значение» (см. рисунок 24-2) появятся данные полученные от прибора. Если же модуль не совместим с устройством, то в поле «Ошибка» (см. рисунок 24-3), напротив всех каналов, появится ошибка «таймаут», т.е. прибор не отвечает. Если ни один модуль не подходит к данному прибору, то придётся разработать его самостоятельно (см. раздел документации 3.1).

2.7.2 Конфигурирование

Как только подходящие модули определены, следует настроить опрос, запустив программу Editor (см. описание 2.1). В первую очередь необходимо выбрать существующую базу конфигурации или создать новую. Для этого, выбрав пункт меню «Файл → Создать/Открыть», можно открыть уже существующую базу, либо вписав желаемое имя в поле «Имя файла:» создать новую конфигурацию. Следующим шагом нужно добавить все требуемые модули опроса в панели подключения модулей устройств (см. описание 2.1.6). Для этого в дереве элементов конфигурации выбрать « Плагины устройств» (см. описание 2.1.2) и нажимая соответствующую кнопку «» (см. рисунок 6-1) выполнить добавление всех необходимых плагинов. Если не выполнить данное действие, то в панели настройки конфигурации прибора (см. описание 2.1.5), будет отсутствовать список плагинов (см. рисунок 5-4). Когда список модулей устройств сформирован, следует добавить все com-порты, через которые будет осуществляться передача данных. Чтобы добавить новый com-порт, необходимо в дереве элементов конфигурации (см. описание 2.1.2) выбрать « Порты / приборы», после чего в панели элементов управления станет активной кнопка добавления нового порта (см. рисунок 3). При нажатии на данную кнопку, появится окно изображённое на рисунке 26, где можно задать номер com-порта.

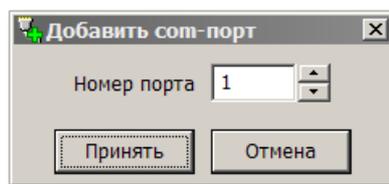


Рисунок 26 – Окно добавления com-порта

После нажатия кнопки «Принять», в дереве элементов конфигурации (см. описание 2.1.2), добавится элемент с названием «com» и номером указанным при создании – « com №». Кроме этого, появится панель настройки com-порта (см. описание 2.1.4), где следует задать все требуемые настроечные параметры. Номер порта всегда можно поменять, задав новое значение в поле «СОМ порт» (см. рисунок 4-3). Как только com-порт настроен, нужно нажать кнопку «Принять» и приступить к добавлению описания прибора (или приборов), который планируется опрашивать. Для этого необходимо в панели элементов управления (см. рисунок 3) нажать соответствующую кнопку. Кнопка будет активна только в том случае, если в дереве элементов конфигурации выбран com-порт. В результате появится окно изображённое на рисунке 27, где нужно задать номер прибора.

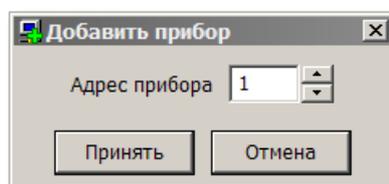


Рисунок 27 – Окно добавления прибора

После того, как будет нажата кнопка «Принять», в дереве элементов конфигурации (см. описание 2.1.2), создастся элемент с номером, указанным при добавлении нового прибора – « №». При этом, появится панель настройки прибора (см. описание 2.1.5), где детально можно задать все необходимые параметры. «Тех. номер» и «Тех. имя» указывать не обязательно. Тут же следует добавить все каналы, которые необходимо будет опрашивать.

Обязательным полем для заполнения является номер канала (см. рисунок 5-9), остальные же параметры являются не обязательными. Как только все данные введены следует нажать на кнопку «Принять».

Последнее что нужно сделать в конфигурировании, это указать модуль (или модули) экспорта данных. При тестовой настройке и использовании программы ReaderGUI (см. описание 2.2), можно и не подключать модули экспорта, а результат опроса наблюдать в самой программе⁸. Для подключения модуля, следует в дереве элементов управления (см. описание 2.1.2) выбрать « Плагины экспорта». И добавив с помощью соответствующей кнопки «» (см. рисунок 7-1) требуемый модуль, задать интервал выполнения в секундах (см. рисунок 7-7). Если ни один модуль не подходит для задачи, то можно самостоятельно разработать такой плагин (см. раздел документации 3.2).

2.7.3 Опрос

Как только конфигурационная база создана, необходимо добавить её имя в файл Reader.ini (см. описание 2.2.5). Если данный файл отсутствует, то нужно запустить программу ReaderGUI (см. описание 2.2), которая создаст требуемый файл. Имя конфигурации добавляется в секцию general в поле database с учётом относительных директорий (см. листинг 1).

После явного указания конфигурации опроса, следует заново запустить ReaderGUI, и уже используя требуемую конфигурацию, запустить опрос устройств, нажав на соответствующую кнопку.

В том случае, если опрос программой ReaderGUI показал удовлетворительный результат, то рекомендуется зарегистрировать опрос как сервис. Для этого следует запустить программу ReaderSvcCtrl (см. описание 2.4), и выбрать пункт 1 – «Установить». Теперь опрос будет осуществляться автоматически после загрузки операционной системы. Если необходимо сразу запустить опрос, то требуется дополнительно выбрать пункт 3 – «Старт».

⁸На данном этапе развития проекта в ПК входит всего лишь один модуль экспорта – экспорт в текстовый файл

3 Руководство разработчика

Обобщённая архитектура работы программы опроса показана на рисунке 28. Как видно из рисунка, ядро программы, только лишь вызывает функции находящиеся в модуле (плагине) и пересылает данные. Ядром программы опроса является файл ReaderCore.dll. Программы ReaderGUI и ReaderSvc используют указанное ядро⁹.

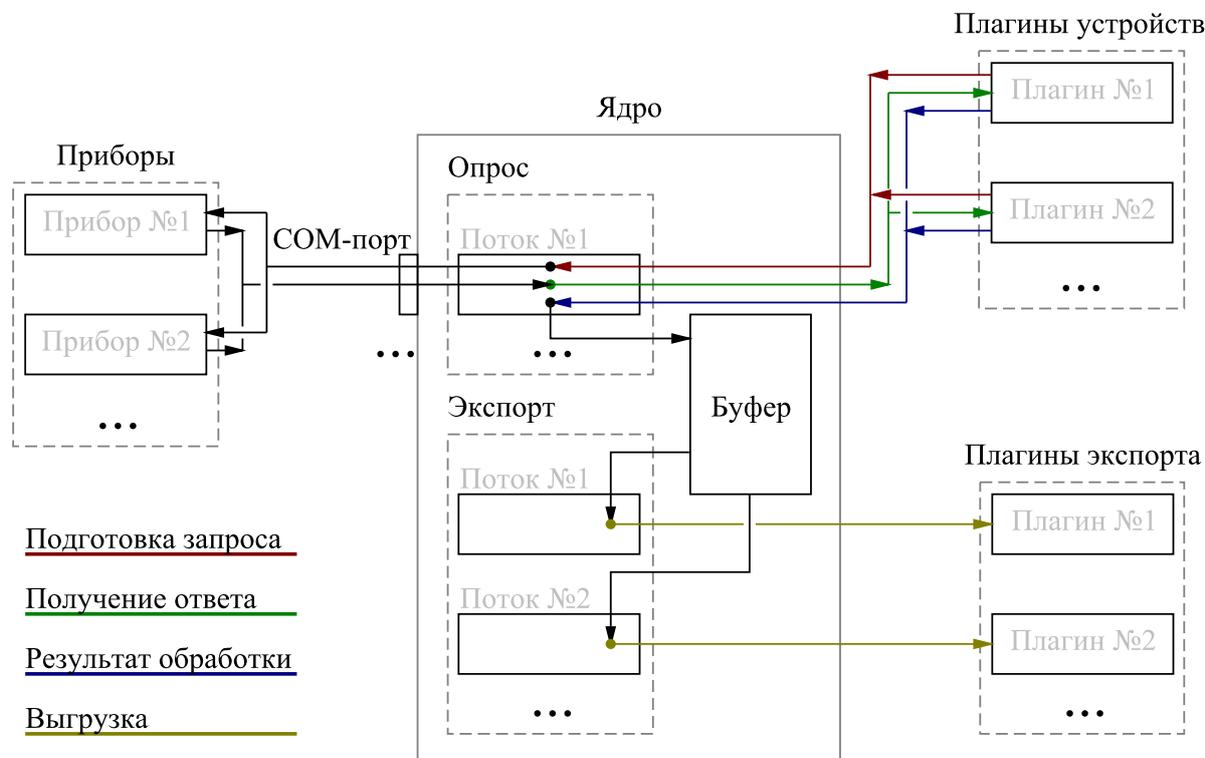


Рисунок 28 – Обобщённая архитектура работы программ опроса

3.1 Модуль (плагин) устройства

Модуль (плагин) в секции экспорта должен содержать 3 функции:

1. `GetInfo` – возвращает информацию о модуле (см. описание 3.1.1).
2. `GetPackage` – формирование пакета (см. описание 3.1.2).
3. `GetData` – обработка пакета (см. описание 3.1.3).

Все функции должны использовать модель вызова **cdecl**, т.е. аргументы передаются через стек, справа налево. Очистку стека производит вызывающая программа.

Пример рабочего исходного кода представлен в описании 3.3.2.

3.1.1 Функция `GetInfo`

Функция `GetInfo` возвращает информацию о плагине. Она не принимает ни одного параметра и описывается следующим образом:

```
const TPluginDeviceInfo* GetInfo();
```

⁹На данном этапе развития проекта ядро никак не подписывается и номер версии узнать нельзя

Функция получает в качестве аргументов:

- package – пакет полученный от прибора. Формальный объём памяти равен inlength (см. листинг 2);
- length – фактическая длина полученного пакета;
- address – адрес устройства от которого ожидался ответ;
- result – результирующие значения. Объём памяти зависит, от типа опроса. Если опрос поканальный, то памяти будет выделено только под одно значение. Если же опрос общий, то памяти будет выделено исходя из номера максимального канала; Структура TChannelInfo описана в листинге 5;
- channels – Если опрос поканальный, то передаётся номер опрошенного канала, иначе передаётся количество опрошенных каналов.

Функция возвращает номер ошибки обработки пакета, либо ошибку полученную от самого прибора (см. таблицу 17).

Листинг 5 – Структура TChannelInfo

```
typedef struct
{
    float value;           // Значение
    unsigned short error; // Ошибка
} TChannelInfo;
```

3.2 Модуль (плагин) экспорта

Модуль (плагин) в секции экспорта должен содержать 4 функции:

1. About – возвращает информацию о модуле (см. описание 3.2.1).
2. Begin – выполняется при старте программы(опроса) (см. описание 3.2.2).
3. End – выполняется при завершении программы(опроса) (см. описание 3.2.2).
4. Export – экспорт данных (см. описание 3.2.3).

Все функции должны использовать модель вызова cdecl, т.е. аргументы передаются через стек, справа налево. Очистку стека производит вызывающая программа.

Пример рабочего исходного кода представлен в описании 3.3.3.

3.2.1 Функция About

Функция About возвращает информацию о плагине. Она не принимает ни одного параметра и описывается следующим образом:

```
const TPluginExportInfo* About();
```

Описание структуры TPluginExportInfo представлено в листинге 6.

Листинг 6 – Структура TPluginExportInfo

```
typedef struct
{
    char* about;    // Название
    char* version; // Версия
    char* author;  // Автор
    char* details; // Детали
} TPluginExportInfo;
```

Все строковые переменные должны заканчиваться нулевым символом.

3.2.2 Функции Begin и End

Функции Begin и End выполняются соответственно при старте и завершении работы опроса. Они не принимают никаких параметров и возвращают 0, в том случае, если выполнение прошло успешно, иначе возвращают значение отличное от нуля. Данные функции необходимы, если, к примеру, необходимо осуществить подключение к БД во время запуска программы и отключение от БД во время её завершения. Если функция Begin не вернёт 0, то функция Export (см. описание 3.2.3) выполняться не будет.

3.2.3 Функции Export

Функции Export выполняет выгрузку данных. Она описывается следующим образом (см. листинг 7):

Листинг 7 – Описание функции Export

```
unsigned short Export(TDataForExport* data, // Список выгружаемых данных
                    unsigned int count);  // Колличество выгружаемых
                                         // данных в списке
```

При удачном выполнении функция должна вернуть значение 0. Структура данных TDataForExport описана в листинге 8.

Листинг 8 – Структура TDataForExport

```
typedef struct
{
    unsigned short port;    // Номер сот-порта
    unsigned short address; // Адрес прибора
    unsigned short channel; // Номер канала

    unsigned int id;        // ID канала (технологический ID)
    char name[16];         // Технологическое имя

    float value;           // Значение
    unsigned short error;  // Ошибка
    time_t time;          // Время считывания
} TDataForExport;
```

Назначение каждого из полей структуры TDataForExport следующее:

- port – номер ком порта через который опрашивается прибор;
- address – адрес опрашиваемого прибора;
- channel – номер опрашиваемого канала;
- id – технологический номер позиции. При отсутствии передаётся 0;
- name – технологическое имя позиции. Строка должна заканчиваться ноль-символом;

- value – значение полученное при опросе данного канала;
- error – ошибка, возникшая при опросе. При положительном исходе имеет значение 0. Номера ошибок см. в таблице 17;
- time – время опроса данного канала. Время представляется по стандарту POSIX, т.е. количество секунд от 1 января 1970 года.

3.3 Дополнительная информация

3.3.1 Номера ошибок

Таблица 17 – Номера возможных ошибок

№ ошибки	Описание
0	Успешное выполнение.
1	Ошибка загрузки плагина.
2	Ошибка открытия com-порта.
3	Ошибка записи в com-порт.
4	Ошибка чтения из com-порта.
5	Ошибка чтения данных (таймаут).
6	Ошибка разбора полученного пакета.
7	Ошибка проверки контрольной суммы.
8	Получен пакет от другого устройства.
9	Канал не настроен.
10	Датчик повреждён.
11	Обрыв связи.
12	Выход за диапазон значений датчика.
13	Прибор повреждён.
20	Смещение для личных (дополнительных) ошибок приборов.

3.3.2 Пример модуля устройства

Пример состоит из двух файлов: заголовочного файла device.h и программы модуля dx2020.c. В файле device.h (листинг 9) описаны все типы данных и определены макросы. В файле dx2020.c (листинг 10) реализация модуля работы с прибором – Yokogawa DX2020.

Листинг 9 – device.h

```
#ifndef DEVICE_H_INCLUDED
#define DEVICE_H_INCLUDED

// Ошибки обработки пакетов
#define OK 0 // Успешное выполнение
#define ERR_PLUGIN_LOAD 1 // Ошибка загрузки плагина
#define ERR_PORT_OPEN 2 // Ошибка открытия com-порта
#define ERR_PORT_WRITE 3 // Ошибка записи в com-порт
#define ERR_PORT_READ 4 // Ошибка чтения из com-порта
#define ERR_READ_DATA_TIMEOUT 5 // Ошибка чтения данных (таймаут)
#define ERR_PARSE_PACKAGE 6 // Ошибка разбора полученного пакета
#define ERR_CRC_CHECK 7 // Ошибка проверки контрольной суммы
#define ERR_INVALID_ADDRESS 8 // Получен пакет от другого устройства
#define ERR_CHANNEL_NOT_CONF 9 // Канал не настроен
#define ERR_SENSOR_FAILURE 10 // Повреждение датчика
#define ERR_SENSOR_DISCONNECT 11 // Обрыв связи
#define ERR_OVERDRAW_LIMIT 12 // Выход за диапазон значений датчика
#define ERR_DEVICE_DAMAGED 13 // Прибор повреждён
```

```

#define ERR_DEVICE_SELF      20 // Смещение для личных ошибок приборов

#define REQUEST_ONE 0
#define REQUEST_ALL 1

typedef struct
{
    char* about;           // Что за плагин?
    char* version;        // Версия
    char* author;         // Автор
    char* details;        // Детали
    unsigned char request; // Тип опроса: поканальный или общий
    unsigned short outlength; // Максимальная длина отправляемого пакета
    unsigned short inlength; // Максимальная длина получаемого пакета
} TPluginDeviceInfo;

typedef struct
{
    float value;
    unsigned short error;
} TChannelInfo;

#define DLL_EXPORT __declspec(dllexport) __cdecl

#ifdef __cplusplus
extern "C"
{
#endif

DLL_EXPORT const TPluginDeviceInfo* GetInfo();

DLL_EXPORT unsigned GetPackage(unsigned short address,
                               unsigned short channel,
                               char* package);

DLL_EXPORT unsigned short GetData(const char* package,
                                  unsigned length,
                                  unsigned short address,
                                  TChannelInfo* result,
                                  unsigned short channels);

#ifdef __cplusplus
}
#endif

#endif // DEVICE_H_INCLUDED

```

Листинг 10 – dx2020.c

```

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include "device.h"

#define DLL_EXPORT __declspec(dllexport) __cdecl

typedef union
{
    unsigned char byte[2];
    short word;
} pair_t;

typedef union
{
    unsigned char byte[2];
    short word;
} pair_t;

pair_t GetCRC(const char* package, unsigned length);
bool CheckCRC(const char* package, unsigned length);
//-----//

```

```

const TPluginDeviceInfo* GetInfo()
{
    static TPluginDeviceInfo info =
    {
        "dx2020",
        "0.1",
        "Андрей Мартыненко",
        "Используя протокол modbus плагины собирают данные "
        "с прибора dx2020 фирмы Yokogawa",
        REQUEST_ALL,
        8,
        64
    };
    return &info;
}
//-----//
unsigned GetPackage(unsigned short address,
                   unsigned short channel,
                   char* package)
{
    package[0] = address;
    package[1] = 0x04;
    package[2] = 0x00;
    package[3] = 0x00;
    package[4] = 0x00;
    package[5] = channel;

    pair_t crc = GetCRC(package, 6);

    package[6] = crc.byte[0];
    package[7] = crc.byte[1];

    return 8;
}
//-----//
// Функция обработки пакета. Возвращает номер ошибки.
unsigned short GetData(const char* package,
                     unsigned length,
                     unsigned short address,
                     TChannelInfo* result,
                     unsigned short channels)
{
    if (!CheckCRC(package, length))
    {
        return ERR_CRC_CHECK;
    }

    if (package[0] != address)
    {
        return ERR_INVALID_ADDRESS;
    }

    if (package[1] != 0x04)
    {
        return ERR_DEVICE_SELF + package[1];
    }

    length = package[2] / 2;

    package += 3;

    pair_t value;

    while (length--)
    {
        value.byte[1] = package[0];
        value.byte[0] = package[1];

        if ((value.word == 0x7fff) || (value.word == 0xffff))
        {
            (*result).error = ERR_CHANNEL_NOT_CONF;
        }
        else

```

```

    {
        (*result).value = value.word;
        (*result).error = 0;
    }

    package += 2;
    result++;
}

return 0;
}
//-----//
pair_t GetCRC(const char* package, unsigned length)
{
    static const char CRCHi[] =
    {
        0x00, 0xc0, 0xc1, 0x01, 0xc3, 0x03, 0x02, 0xc2, 0xc6, 0x06, 0x07, 0xc7, 0x05, 0xc5,
        0xc4, 0x04, 0xcc, 0x0c, 0x0d, 0xcd, 0x0f, 0xcf, 0xce, 0x0e, 0x0a, 0xca, 0xcb, 0x0b,
        0xc9, 0x09, 0x08, 0xc8, 0xd8, 0x18, 0x19, 0xd9, 0x1b, 0xdb, 0xda, 0x1a, 0x1e, 0xde,
        0xdf, 0x1f, 0xdd, 0x1d, 0x1c, 0xdc, 0x14, 0xd4, 0xd5, 0x15, 0xd7, 0x17, 0x16, 0xd6,
        0xd2, 0x12, 0x13, 0xd3, 0x11, 0xd1, 0xd0, 0x10, 0xf0, 0x30, 0x31, 0xf1, 0x33, 0xf3,
        0xf2, 0x32, 0x36, 0xf6, 0xf7, 0x37, 0xf5, 0x35, 0x34, 0xf4, 0x3c, 0xfc, 0xfd, 0x3d,
        0xff, 0x3f, 0x3e, 0xfe, 0xfa, 0x3a, 0x3b, 0xfb, 0x39, 0xf9, 0xf8, 0x38, 0x28, 0xe8,
        0xe9, 0x29, 0xeb, 0x2b, 0x2a, 0xea, 0xee, 0x2e, 0x2f, 0xef, 0x2d, 0xed, 0xec, 0x2c,
        0xe4, 0x24, 0x25, 0xe5, 0x27, 0xe7, 0xe6, 0x26, 0x22, 0xe2, 0xe3, 0x23, 0xe1, 0x21,
        0x20, 0xe0, 0xa0, 0x60, 0x61, 0xa1, 0x63, 0xa3, 0xa2, 0x62, 0x66, 0xa6, 0xa7, 0x67,
        0xa5, 0x65, 0x64, 0xa4, 0x6c, 0xac, 0xad, 0x6d, 0xaf, 0x6f, 0x6e, 0xae, 0xaa, 0x6a,
        0x6b, 0xab, 0x69, 0xa9, 0xa8, 0x68, 0x78, 0xb8, 0xb9, 0x79, 0xbb, 0x7b, 0x7a, 0xba,
        0xbe, 0x7e, 0x7f, 0xbf, 0x7d, 0xbd, 0xbc, 0x7c, 0xb4, 0x74, 0x75, 0xb5, 0x77, 0xb7,
        0xb6, 0x76, 0x72, 0xb2, 0xb3, 0x73, 0xb1, 0x71, 0x70, 0xb0, 0x50, 0x90, 0x91, 0x51,
        0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9c, 0x5c,
        0x5d, 0x9d, 0x5f, 0x9f, 0x9e, 0x5e, 0x5a, 0x9a, 0x9b, 0x5b, 0x99, 0x59, 0x58, 0x98,
        0x88, 0x48, 0x49, 0x89, 0x4b, 0x8b, 0x8a, 0x4a, 0x4e, 0x8e, 0x8f, 0x4f, 0x8d, 0x4d,
        0x4c, 0x8c, 0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
        0x41, 0x81, 0x80, 0x40
    };

};

static const unsigned char CRCLo[] =
{
    0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1,
    0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x00, 0xc1, 0x81, 0x40,
    0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0,
    0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41,
    0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x00, 0xc1,
    0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40,
    0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0,
    0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41,
    0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0,
    0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41,
    0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0,
    0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41,
    0x00, 0xc1, 0x81, 0x40
};

pair_t crc;

crc.word = 0xffff;

while (length--)
{
    unsigned index = crc.byte[0] ^ (unsigned char) *package++;
    crc.byte[0] = crc.byte[1] ^ CRCLo[index];
    crc.byte[1] = CRCHi[index];
}

return crc;
}
//-----//

```

```
bool CheckCRC(const char* package, unsigned length)
{
    if (length < 3)
    {
        return false;
    }

    pair_t crc;

    crc.byte[0] = package[length-2];
    crc.byte[1] = package[length-1];

    return crc.word == GetCRC(package, length-2).word;
}
```

3.3.3 Пример модуля экспорта

Пример состоит из двух файлов: заголовочного файла `export.h` и программы модуля `text.c`. В файле `export.h` (листинг 11) описаны все типы данных и определены макросы. В файле `text.c` (листинг 12) реализация модуля выгрузки данных в текстовый файл.

Листинг 11 – `export.h`

```
#ifndef EXPORT_H_INCLUDED
#define EXPORT_H_INCLUDED
#include <time.h>

// Ошибки обработки пакетов
#define OK 0 // Успешное выполнение
#define ERR_PLUGIN_LOAD 1 // Ошибка загрузки плагина
#define ERR_PORT_OPEN 2 // Ошибка открытия com-порта
#define ERR_PORT_WRITE 3 // Ошибка записи в com-порт
#define ERR_PORT_READ 4 // Ошибка чтения из com-порта
#define ERR_READ_DATA_TIMEOUT 5 // Ошибка чтения данных (таймаут)
#define ERR_PARSE_PACKAGE 6 // Ошибка разбора полученного пакета
#define ERR_CRC_CHECK 7 // Ошибка проверки контрольной суммы
#define ERR_INVALID_ADDRESS 8 // Получен пакет от другого устройства
#define ERR_CHANNEL_NOT_CONF 9 // Канал не настроен
#define ERR_SENSOR_FAILURE 10 // Повреждение датчика
#define ERR_SENSOR_DISCONNECT 11 // Обрыв связи
#define ERR_OVERDRAW_LIMIT 12 // Выход за диапазон значений датчика
#define ERR_DEVICE_DAMAGED 13 // Прибор повреждён

#define ERR_DEVICE_SELF 20 // Смещение для личных ошибок приборов

static const char* CListErrors[] = { "",
    "Ошибка загрузки плагина",
    "Ошибка открытия com-порта",
    "Ошибка записи в com-порт",
    "Ошибка чтения из com-порта",
    "Таймаут",
    "Ошибка разбора полученного пакета",
    "Ошибка проверки контрольной суммы",
    "Получен пакет от другого устройства",
    "Канал не настроен",
    "Повреждение датчика",
    "Обрыв связи",
    "Выход за диапазон значений датчика",
    "Прибор повреждён" };

static const unsigned CCountListErrors = sizeof(CListErrors) /
    sizeof(*CListErrors);

#define TECH_NAME_LENGTH 16

typedef struct
{
    unsigned short port; // Номер com-порта
    unsigned short address; // Адрес прибора
}
```

```

    unsigned short channel;           // Номер канала

    unsigned int id;                 // ID канала (технологический ID)
    char name[TECH_NAME_LENGTH];     // Технологическое имя

    float value;                     // Значение
    unsigned short error;             // Ошибка
    time_t time;                     // Время считывания
} TDataForExport;

typedef struct
{
    char* about; // Что за плагин?
    char* version; // Версия
    char* author; // Автор
    char* details; // Детали
} TPluginExportInfo;

#define DLL_EXPORT __declspec(dllexport) __cdecl

#ifdef __cplusplus
extern "C"
{
#endif

DLL_EXPORT const TPluginExportInfo* About();

DLL_EXPORT unsigned short Begin();

DLL_EXPORT unsigned short End();

DLL_EXPORT unsigned short Export(TDataForExport *data,
                                unsigned int count);

#ifdef __cplusplus
}
#endif

#endif // EXPORT_H_INCLUDED

```

Листинг 12 – text.c

```

#include <stdio.h>
#include "export.h"

#define FILENAME "values.txt"

static FILE* f = NULL;

//-----//
const TPluginExportInfo* About()
{
    static TPluginExportInfo info=
    {
        "textfile",
        "0.1",
        "Андрей Мартыненко",
        "Плагин рассчитан на выгрузку данных в текстовый файл"
    };
    return &info;
};
//-----//
unsigned short Begin()
{
    f = fopen(FILENAME, "wb");
    return !f;
}
//-----//
unsigned short End()
{
    if (f)

```

```
{
    fclose(f);
}
return 0;
}
//-----//
unsigned short Export (TDataForExport* data,
                      unsigned int count)
{
    if (!f)
    {
        return 1;
    }

    char date[20];
    memset(date, 0, sizeof(date));

    unsigned int i;

    for (i=0; i<count; ++i)
    {
        struct tm* ptr = localtime(&(amp;data[i].time));
        strftime(date, sizeof(date)-1, "%d.%m.%y %H:%M:%S", ptr);

        fprintf(f, "%3d | %3d | %3d | %6d | %10.3f | %36s | %s\n",
                data[i].port,
                data[i].address,
                data[i].channel,
                data[i].id,
                data[i].value,
                data[i].error < CCountListErrors ? CListErrors[data[i].error] :
                "Неизвестная ошибка",
                date);
    }
    fseek(f, 0, SEEK_SET);
    return 0;
}
```

3.3.4 Инструментальные средства использованные в проекте

Компиляторы: GNU GCC, Tiny C Compiler
Библиотеки: wxWidgets, SQLite
Среда разработки: Code::Blocks
Редактор форм: wxFormBuilder