**АННОТАЦИЯ**

Выпускная квалификационная работа магистра на тему: «Разработка автоматизированной системы контроля и проверки оборудования» описывает процесс создания программы контроля работоспособности (далее по тексту CheckFut).

Раздел «Специальная часть» включает в себя постановку задачи, обоснование выбора языка и среды разработки, сравнительный анализ существующих решений в соответствии с выделенным критериями, алгоритм решения задачи, описание структуры входных и выходных данных, программные и аппаратные требования, систему и методику тестирования программного продукта.

Раздел «Заключение» содержит выводы о проделанной работе в процессе разработки.

«Список использованной литературы» соответствует использованным при разработке проекта источникам информации.

Текст программы и слайды презентации представлены в приложениях.

**Содержание**

[ВВЕДЕНИЕ 5](#_Toc31097570)

[1. СПЕЦИАЛЬНАЯ ЧАСТЬ 6](#_Toc31097571)

[1.1. Постановка задачи 6](#_Toc31097572)

[1.2. Обзор и анализ существующих решений 6](#_Toc31097573)

[1.3. Описание алгоритма 14](#_Toc31097574)

[1.4. Организация данных 29](#_Toc31097575)

[1.4.1. Организация входных данных 29](#_Toc31097576)

[1.4.2. Организация выходных данных 30](#_Toc31097577)

[1.5. Требования к составу и параметрам технических средств 30](#_Toc31097578)

[1.6. Требования к информационной и программной совместимости 30](#_Toc31097579)

[2. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ И ИХ ОЦЕНКА 31](#_Toc31097580)

[2.1. Объект испытаний 31](#_Toc31097581)

[2.2. Цель испытаний 31](#_Toc31097582)

[2.3. Требования к программе 31](#_Toc31097583)

[2.4. Средства и порядок проведения испытаний 31](#_Toc31097584)

[2.4.1. Технические средства, используемые во время испытаний 31](#_Toc31097585)

[2.5.2. Программные средства, используемые во время испытаний 32](#_Toc31097586)

[2.6. Методы испытаний 32](#_Toc31097587)

[3. ОПИСАНИЕ ОКОН И ЭЛЕМЕНТОВ ПРОГРАММЫ 38](#_Toc31097588)

[3.1. Описание главного окна программы 38](#_Toc31097589)

[3.2. Описание вспомогательных окон программы 42](#_Toc31097590)

[ЗАКЛЮЧЕНИЕ 50](#_Toc31097591)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 51](#_Toc31097592)

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ 52](#_Toc31097593)

[ПРИЛОЖЕНИЯ 54](#_Toc31097594)

[Приложение А. Текст программы 55](#_Toc31097595)

[Приложение Б. Формат протокола 56](#_Toc31097596)

[Приложение В. Акт о внедрении 80](#_Toc31097597)

# ВВЕДЕНИЕ

Задача создания автоматизированной контрольно-проверочной аппаратуры (АКПА) состоит из разработки аппаратной части и программного обеспечения. Аппаратной составляющей является стойка с измерительными приборами, разработанная на предприятии ООО НПП «Редан». В данной выпускной квалификационной работе рассматривается процесс разработки программного продукта для работы как на данной стойке, так и на любом ПК, совместимым с требуемыми параметрами. Объектом контроля является модуль аппаратурный (МА) подводного аппарата. МА состоит из нескольких программируемых логических интегральных схем с процессорами 1967ВН028 производства российской компании «Миландр», на которые загружаются бортовые программы. Именно на обмене данными с этими программами строится система контроля.

# 1. СПЕЦИАЛЬНАЯ ЧАСТЬ

# 1.1. Постановка задачи

Задача: обеспечить автоматизированный контроль работоспособности с помощью АКПА на предприятии ООО НПП «Редан».

Исходные данные:

* стойка, разработанная на предприятии, в состав которой входят: преобразователь Advantech EKI-1528, коммутатор LAN MOXA PowerTrans PT-7728, четыре источника постоянного тока TDK LAMBDA GEN-40-19, генератор сигналов АКИП 3402, кондиционер сетевого питания Furman PL-8C-E, многофункциональный блок Agilent 34980A, мультиметр FLUKE 8846A, калибратор давления Mensor CPC600;
* документация на приборы;
* Ethernet-кабель для подключения ПК к стойке;
* МА в собранном состоянии.

Требуется:

* разработать программы для проверки всех подсистем МА;
* разработать программу-менеджер, которая организует запуск проверок и контроль их выполнения;
* разработать библиотеку для работы с приборами, входящими в стойку, продумать механизм обмена данными с программой-менеджером;
* разработать надстройку на открытую библиотеку LibHaru для эффективного формирования протокола контроля в формате .pdf.

# 1.2. Обзор и анализ существующих решений

Для решения поставленной задачи и удовлетворения всех требований рассматривалась возможность применения готовых программ, которые не подошли по тем или иным причинам. Системы оценивались по следующим критериям:

1) работа как в ручном, так и в автоматизированном режиме;

2) контроль всех подсистем МА;

3) высокая модифицируемость программы в рамках отдельных проверок;

4) создание защищенного протокола работы.

**1.2.1. Работа как в ручном, так и в автоматизированном режиме**

Система должна обеспечивать контроль работоспособности в автоматизированном режиме с минимальным участием оператора. Это один из основных параметров, который обеспечивает более высокую скорость проверки вместе с возможностью параллельного запуска нескольких экземпляров ПО на одной стойке. Также необходима функция проведения ручного контроля для детального выявления неисправностей при ошибках в автоматизированном режиме. Этот вид контроля должен предусматривать проверки отдельных подсистем с различными параметрами ввода.

**1.2.2. Контроль всех подсистем МА**

Этот критерий характеризует полноту проверок модуля как на аппаратном, так и на прикладном уровне. Набор измерительных приборов в стойке должен обеспечивать возможность полноценного контроля, а ПО его реализовывать.

**1.2.3. Высокая модифицируемость программы в рамках отдельных проверок**

Так как в программу могут добавляться новые проверки и изменяться логика старых, то программа должна обладать высокой степенью модифицируемости. Также необходим функционал для выявления неисправностей в случаях ошибки автоматизированного контроля.

**1.2.4. Создание защищенного протокола работы**

Программно-аппаратный комплекс планируется поставлять различным контрагентам и на испытания, в следствие чего необходимо предусмотреть создание отчета по результатам работы программы. Такой отчет должен быть защищен от изменения для целесообразности дальнейшего анализа.

Рассматривались следующие программно-аппаратные комплексы:

1. Автоматизированный комплекс для регулировки и проведения приемо-сдаточных испытаний модуля интерфейса подводного аппарата.

2. Программа проверки работоспособности модуля аппаратурного «Tkpa.exe».

**1.2.5. Автоматизированный комплекс для регулировки и проведения приемо-сдаточных испытаний модуля интерфейса подводного аппарата.**

Данный автоматизированный комплекс разработан на предприятии АО «Ижевский радиозавод», специализирующийся в основном на разработке контрольно-проверочной аппаратуры для космических аппаратов. Это предприятие реализовывает как аппаратную, так и программную часть, и принимает заказы на изготовление стоек с готовым ПО.

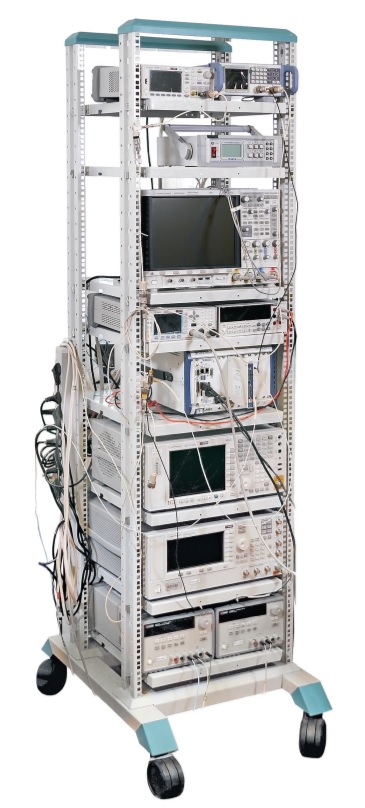


Рис. 1.1 - Внешний вид стойки ИРЗ

В состав данного комплекса входят:

1. Блоки сопряжения с объектом контроля.

2. Осциллограф.

3. Измеритель частот.

4. Аттенюатор.

5. Анализатор спектра.

6. Генератор сигнала.

7. Приемопередатчик векторных сигналов.

8. Программируемый источник питания.

9. Программное обеспечение АРМ оператора.

ПО разработано на базе среды графического программирования LabVIEW и обладает следующим функционалом:

1. Создание сценариев испытаний объектов контроля с помощью среды NI TestStand.

2. Протоколирование работы.

3. Управление питанием.

4. Управление передатчиком и приемником изделия[1].

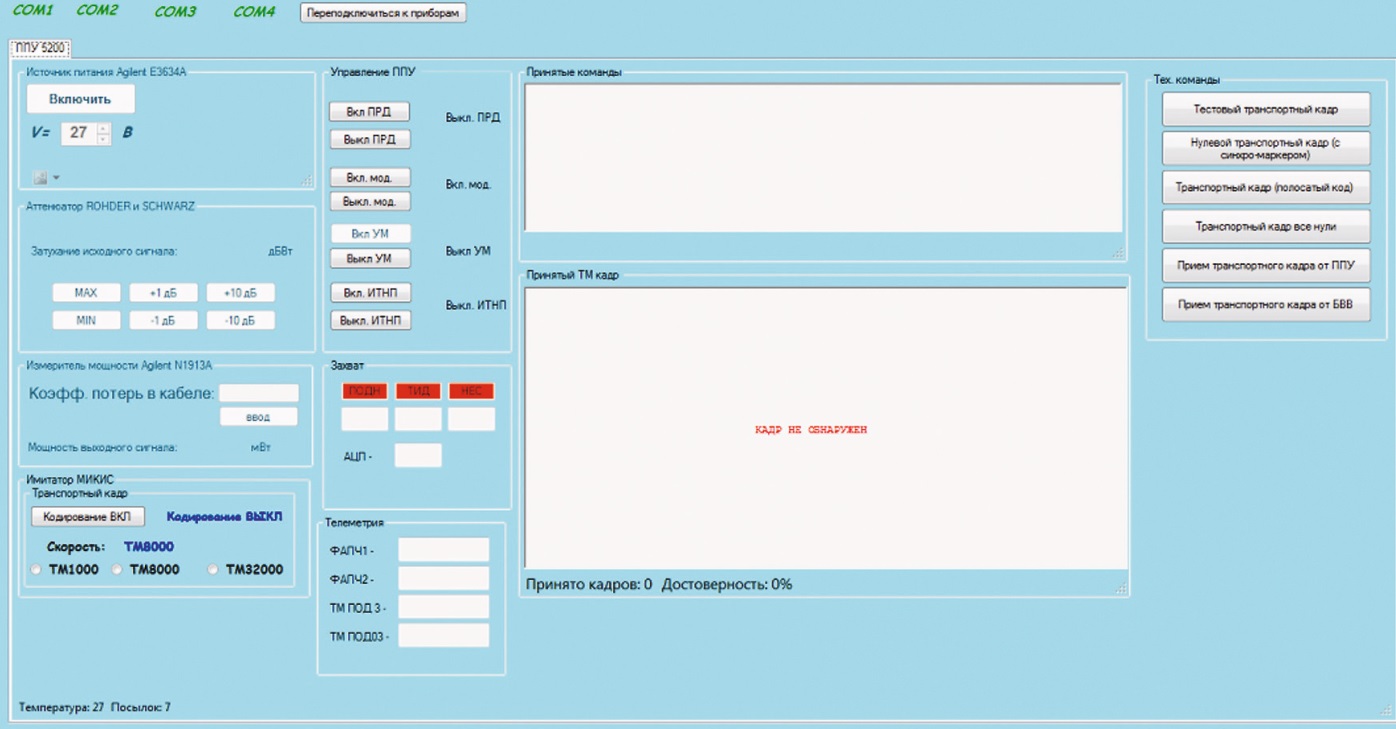


Рис. 1.2 - Интерфейс программы для регулировки и проведения приемо-сдаточных испытаний модуля интерфейса подводного аппарата

Программа предоставляет возможность как автоматизированного, так и ручного контроля. Возможен контроль отдельных подсистем, ввод различных параметров, включая напряжение, частоту исходного сигнала, коэффициент потерь и т.д. Также программа отображает последовательность битов в принимаемом кадре, что облегчает поиск ошибок в обмене командами.

Так как, данный комплекс является сторонней разработкой, то можно выявить ряд недостатков. Например, комплекс не способен обеспечить полноту проверок МА из-за отсутствия таких приборов как мультиметр и калибратор давления. Также нет возможности добавления и изменения проверок, так как это законченный комплекс без подобного функционала. В этом случае добавление новых возможностей и даже однотипных функций влечет за собой взаимодействие с разработчиком ПО, согласованием и сдачей готового продукта. Протокол работы формируется в формате .txt, что также является недостатком, так как подобные комплексы планируется отправлять контрагентам ООО НПП «Редан», после чего получать от них результаты работы. Невозможно доказать достоверность протокола, содержимое которого можно изменить извне программы.

**1.2.6. Программа проверки работоспособности модуля аппаратурного «Tkpa.exe»**

Программа «Tkpa.exe» разработана на предприятии ООО НПП «Редан» и используется для контроля работоспособности МА и формирования протокола работы по каждой проверке. Данная программа написана в 2010 году, дополнялась и изменялась множество раз, но альтернатив ей на предприятии до сих пор нет.

Программа предоставляет возможность как автоматизированной проверки, так и ручного контроля. В обоих режимах доступна и полная, и частичная проверка. Кроме того, структура программы предусматривает запуск отдельных тестов в двух режимах: усеченном и расширенном. Усеченный вид запускается из-под программы-менеджера (есть возможность запустить и отдельно) и проводит проверку в автоматическом режиме согласно внутренней логике программы и заранее определённым входным параметрам. Расширенный режим можно запустить только вручную. Он в основном используется для диагностики неполадок и позволяет детально настроить входные данные, а также в реальном времени отслеживать выходные параметры.

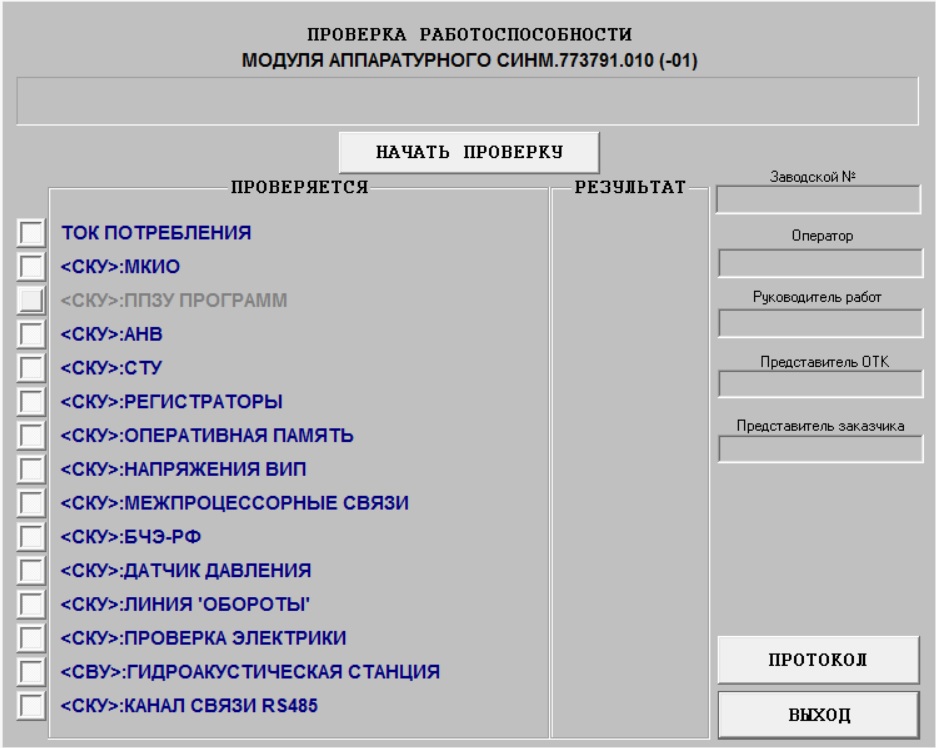


Рис. 1.3 - Интерфейс программы «Tkpa.exe»

Программа позволяет проверить все подсистемы МА, но без возможности аппаратного управления приборами, их приходится включать, выключать и настраивать вручную. Так как программа написана на предприятии, то также имеется исходный код с возможностью его модификации. Это позволяет убирать, или наоборот, добавлять проверки, корректировать их, менять порядок и прочее.

Также есть возможность получения и сохранения протокола по результатам проделанной работы. По каждому тесту формируются отдельные файлы-отчеты, которые в дальнейшем объединяются в единый протокол. Однако форматом файла является .txt, что является серьезным недостатком, так как его можно изменить извне программы.

Но основным недостатком данной программы, конечно, является отсутствие автоматизации. Вместо управления приборами их приходится подключать вручную во время появления соответствующий информационных окон. Сводные данные о соответствии рассмотренных программных средств выбранным критериям приводится в таблице 1.1.

Таблица 1.1 - Сводные данные о соответствии рассмотренных ПС критериям

|  |  |  |  |
| --- | --- | --- | --- |
|  | Комплекс «ИРЗ» | Программа проверки работоспособности модуля аппаратурного | Комплект программ контроля технического состояния блока электронного |
| Ручной режим | + | + | + |
| Автоматизированный режим | + | - | + |
| Контроль всех подсистем МА | - | + | + |
| Высокая модифицируемость | - | + | + |
| Защищенный протокол | - | - | + |

В результате проведенного анализа подтверждена необходимость разработки нового комплекса программ, который, в соответствии с выбранными критериями, должен:

* запускать проверки как в ручном, так и в автоматизированном режиме;
* обеспечивать на аппаратном и программном уровне контроль всех подсистем модуля аппаратурного и блока электронного;
* иметь высокую степень модифицируемости как в рамках отдельных тестов, так и рамках главной программы-менеджера;
* сохранять результаты работы в защищенный файл отчета в формате .pdf.

**1.2.7. Выбор языка и среды разработки**

По требованию заказчика исходные коды программы должны быть реализованы на языке С++ с использованием библиотек MFC, среди преимуществ которого можно выделить следующие:

* сочетание высокоуровневых и низкоуровневых средств;
* реализация ООП;
* кроссплатформенность;
* высокая совместимость с языком C.

Удобство языка С++ основано на том, что он является языком высокого уровня, имеющим полный набор конструкций структурного программирования, поддерживающим модульность, блочную структуру программ, возможность раздельной компиляции модулей. В то же самое время язык С++ имеет набор низкоуровневых средств, позволяющих иметь удобный доступ к аппаратным средствам компьютера, в частности, позволяющих добраться до каждого бита памяти[2]. В данной работе это особенно важно при реализации алгоритмов обмена информацией с процессорами БЭ.

Главное отличие языка С от С++ – это поддержка объектно-ориентированного программирования. Объектно-ориентированное программирование объединило лучшие идеи структурированного с рядом мощных концепций, которые способствуют более эффективной организации программ. Объектно-ориентированный подход к программированию позволяет разложить задачу на составные части таким образом, что каждая составная часть будет представлять собой самостоятельный объект, который содержит собственные инструкции и данные. При таком подходе существенно понижается общий уровень сложности программ, что позволяет программисту справляться с более сложными программами, чем раньше (т.е. написанными при использовании структурированного программирования). Все языки объектно-ориентированного программирования характеризуются тремя общими признаками: инкапсуляцией, полиморфизмом и наследованием[3]. В данной работе ООП жизненно необходимо для написания оконных приложений-тестов и головной программа-менеджера, работающих в диалоге с пользователем. Программы, написанные с помощью ООП, обладают высокой степенью модифицируемости, что является одним из необходимых параметров.

С++ является кроссплатформенным языком: стандарт накладывает минимальные требования на ЭВМ для запуска скомпилированных программ. На практике, для написания портируемого кода на С++ требуется огромное мастерство и опыт, и «небрежные» коды на С++ с высокой вероятностью могут оказаться непортируемыми. Тонкое владение С++ в принципе может сделать код на С++ столь же портируемым, что и код на С[4]. В данной работе это свойство важно, так как неизвестно на какой ЭВМ будет в дальнейшем дорабатываться проект, и на каких машинах запускаться скомпилированные экземпляры программ.

Развитие языка С++ происходило на базе языка С, и, за небольшим исключением, C был сохранен в качестве подмножества C++. Базовый язык С был спроектирован таким образом, что имеется очень тесная связь между типами, операциями, операторами и объектами, с которыми непосредственно работает машина, т.е. числами, символами и адресами. За исключением операций new, delete и throw, а также проверяемого блока, для выполнения операторов и выражений С++ не требуется скрытой динамической аппаратной или программной поддержки[5]. В данной работе это свойство особенно важно при работе с технологическими программами (написанными на чистом С), которые загружаются в процессоры БЭ.

По требованию заказчика в качестве среды разработки программы должна быть использована Microsoft Visual Studio 2015 Ultimate.

# 1.3. Описание алгоритма

**1.3.1. Общий алгоритм работы**

На рисунке 1.4 представлена схема алгоритма работы программы.



Рис. 1.4 - Схема алгоритма работы программы-менеджера

Программный комплекс состоит из программ-тестов отдельных подсистем МА и программы-менеджера, которая управляет приборами в стойке, запускает тесты, выдает диагностические сообщения, формирует протокол по результатам работы. Подобная схема с разделением проверок на отдельные программы хорошо себя зарекомендовала. В отличие от варианта единой управляющей программой со всеми возможными проверками, выбранная модель обладает высокой отказоустойчивостью и возможностью модификации. В частности, если потребуется изменить или дополнить отдельную проверку, достаточно отладить только один тест, совершенно не затрагивая остальные компоненты программного комплекса.

После запуска программы-менеджера необходимо ввести ФИО оператора, представителя ОТК и ВП, выбрать место проведения работ, тип изделия и хранения (от этого зависит набор и логика нескольких тестов), а также ввести заводской номер.

Кроме того, во время запуска происходит опрос подключенных приборов и обмен с ними тестовыми сообщениями. Если все приборы подключены и исправны, то контроль возможно проводить в автоматическом режиме. Если же с частью приборов нет связи, то ими можно управлять вручную и провести контроль в автоматизированном режиме: пользователь будет включать и выключать приборы при появлении соответствующих окон с подсказками. После нажатия кнопки «Начать контроль» проверяется наличие exe-файлов программ тестов. Если файлов нет, то для продолжения работы предлагается указать эти файлы. Далее идет сам контроль материальной части, по очереди запускаются тесты различных подсистем с выдачей в конце результата как в рамках отдельных проверок, так и окончательного. Также в конце формируется протокол с детальной информацией по каждому тесту.

Информация обо всех проверках содержится в бинарном файле config.dat, который считывается программой-менеджером при запуске. Для каждого теста в файле имеется структура, формат которой представлен в таблице 1.2.

Таблица 1.2 - Структура данных для запуска тестов

|  |  |  |
| --- | --- | --- |
| Тип данных | Название | Комментарий |
| TCHAR [256] | lpszCheckName | Название теста |
| TCHAR [1024] | lpszPath | Путь к исполняемому файлу теста |
| TCHAR [5][1024] | lpszCmdPar | Массив параметров командной строки |
| TCHAR [1024] | lpszRepName | Название файла протокола теста |
| int | nResult | Результат контроля |
| int | iCountRun | Количество запусков теста |
| int | nMode | Режим, в котором доступен тест |
| int | nPrevAction | Действие перед тестом |
| int | nPostAction | Действие после теста |

Чтобы не перегружать головную программу CheckFut, контроль за приборами было решено вынести в отдельную программу DeviceManagment, которая принимает команды от менеджера и посылает их дальше на приборы. Общая схема взаимодействия программ в комплексе представлена на рисунке 1.5.



Рис. 1.5 - Схема взаимодействия программ

**1.3.2. Описание алгоритма работы программы DeviceManagment**

Программа DeviceManagment запускается в свернутом виде перед началом контроля и также может работать как в ручном, так и в автоматическом режиме под управлением головной программы. Обмен данными между программами осуществляется средствами межпроцессорного взаимодействия с общей памятью (Shared Memory) с использованием потоков (CThread), событий (CEvent), блокировок (CMutex) и сообщений (PostMessage). Интерфейс DeviceManagment для ручного режима представлен на рисунке 1.6.

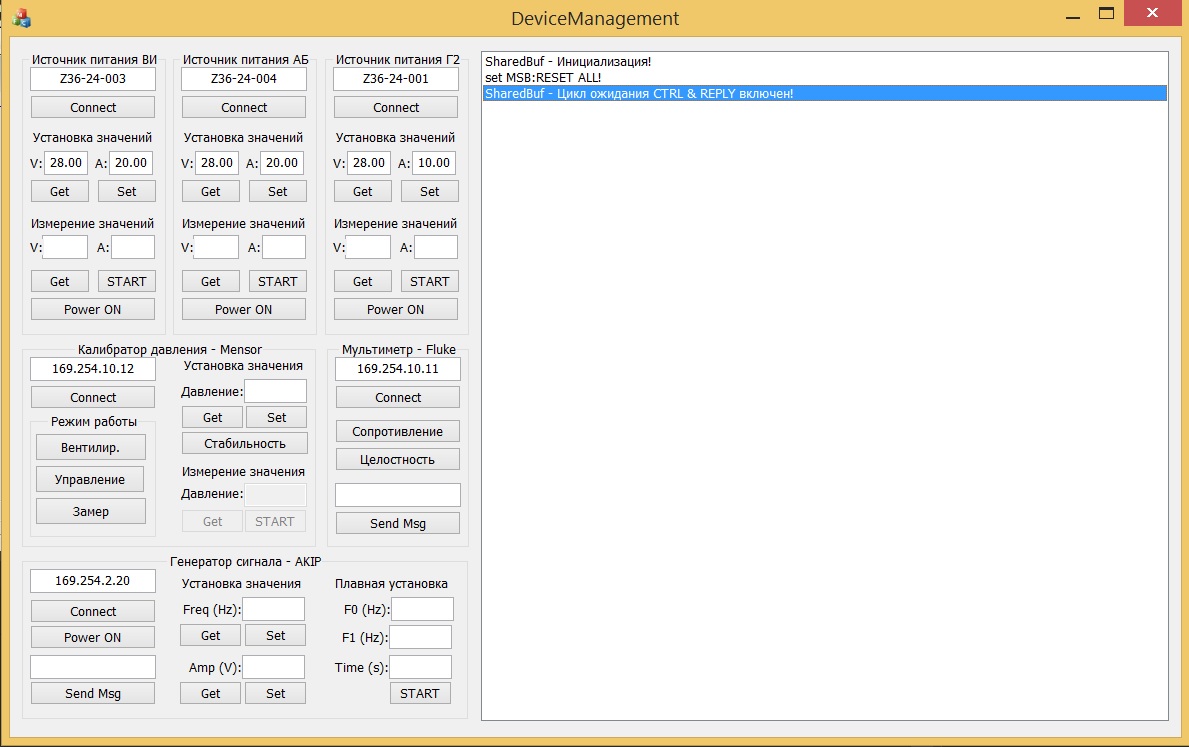


Рис. 1.6 - Интерфейс программы DeviceManagment

Для управления всеми видами приборов был разработан протокол взаимодействия, обеспечивающий достоверность передаваемых данных.

Таблица 1.3 - Общая структура \_SB\_GLOBAL

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование параметра | Тип (размер) | Диапазон, размерность | Примечание |
| Информационное поле | \_SB\_STR\_INFO  (512 байт) |  | «INFO» |
| Управляющие поле | \_SB\_STR\_CONTROL (2048 байт) |  | «CONTROL» |
| Поле ошибок | \_SB\_STR\_ERROR (512 байт) |  | «ERROR» |
| Ответное поле | \_SB\_STR\_REPLY (2048 байт) |  | «REPLY» |
| Резерв | BYTE (4096 байт) |  |  |

Программа «DeviceManagement» осуществляет:

* прием сообщения из поля CONTROL;
* заполнение сообщения в поле INFO;
* заполнение сообщения в поле ERROR;
* заполнение сообщения в поле REPLY.

Программа «CheckFut» осуществляет:

* заполнение сообщения в поле CONTROL;
* прием сообщения из поля INFO;
* прием сообщения из поля ERROR;
* прием сообщения из поля REPLY.

Таблица 1.4 - Структура информационного поля \_SB\_STR\_INFO

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование параметра | Тип (размер) | Диапазон, размерность | Примечание |
| Идентификатор начала | uint32 (4 байта) | 0x4F464E49 | “INFO” |
| Номер версии программы | uint16 (2 байта) | 1 |  |
| Номер версии протокола | uint16 (2 байта) | 1 |  |
| Размер общего буфера | uint16 (2 байта) | 9216 | Размер \_SB\_GLOBAL |
| Количество заполненных статусов устройств | uint8 (1 байт) |  |  |
| Резерв | int8 (93 байта) |  |  |
| Набор статусов устройств | \_SB\_DEV\_STATUS  (400 байт) | 100 шт. | Массив структур |
| Контрольная сумма | uint32 (4 байта) |  | Сумма байт |
| Идентификатор конца | uint32 (4 байта) | 0x494E464F | “OFNI” |

Таблица 1.5 - Структура информационного поля \_SB\_DEV\_STATUS поля \_SB\_STR\_INFO

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование параметра | Тип (размер) | Диапазон, размерность | Примечание |
| Тип устройства | uint8 (1 байт) |  | Начало с 1 |
| Номер устройства | uint8 (1 байт) |  |  |
| Режим работы | uint8 (1 байт) |  |  |
| Состояние питания | uint8 (1 байт) |  |  |

Таблица 1.6 - Структура управляющего поля \_SB\_STR\_CONTROL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Наименование параметра | Тип (размер) | Диапазон, размерность | Примечание |
| 1 | Идентификатор начала | uint32 (4 байта) | 0x4C525443 | “CTRL” |
| 2 | Управляющие команды | BYTE  (2036 байт) | \_STR\_MASK\_CONTROL | Накладывается маска структур |
| 3 | Контрольная сумма | uint32 (4 байта) |  | Сумма байт |
| 4 | Идентификатор конца | uint32 (4 байта) | 0x4354524C | “LRTC” |

Таблица 1.7 - Структура поля ошибок \_SB\_STR\_ERROR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Наименование параметра | Тип (размер) | Диапазон, размерность | Примечание |
| 1 | Идентификатор начала | uint32 (4 байта) | 0x21525245 | “ERR!” |
| 2 | Код устройства | int16 (2 байта) |  |  |
| 3 | Код ошибки | int16 (2 байта) |  |  |
| 4 | Информационная строка | TCHAR  (496 байт) | TCHAR[248] | Строка ошибки |
| 5 | Контрольная сумма | uint32 (4 байта) |  | Сумма байт |
| 6 | Идентификатор конца | uint32 (4 байта) | 0x45525221 | “!RRE” |

Таблица 1.8 - Структура ответного поля \_SB\_STR\_REPLY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Наименование параметра | Тип (размер) | Диапазон, размерность | Примечание |
| 1 | Идентификатор начала | uint32 (4 байта) | 0x4C504552 | “REPL” |
| 2 | Ответная информация | BYTE  (2036 байт) | \_STR\_MASK\_REPLY | Накладывается маска структур |
| 3 | Контрольная сумма | uint32 (4 байта) |  | Сумма байт |
| 4 | Идентификатор конца | uint32 (4 байта) | 0x5245504C | “LPER” |

Программа состоит из 8 классов:

* CControlPanalApp – главный класс программы в структуре MFC. Из него вызывается конструктор диалогового окна;
* CControlPanelDlg – класс основного диалогового окна. Отвечает за отображение информации, позволяет оператору в ручном режиме управлять приборами;
* CSharedBuf – класс для работы с общей памятью. Также используется в программе-менеджере. Позволяет создавать и уничтожать общие области памяти, передавать и получать информацию из них;
* CNetwork – класс, осуществляющий соединение и отправку сообщений по сети TCP на роутер, далее на концентратор и устройства. Позволяет настроить сокеты, отправлять и принимать сообщения;
* CTDK\_LambdaZ – класс для управления источником постоянного тока TDK LAMBDA GEN-40-19. Позволяет соединиться, включить прибор, установить значения выходного тока и напряжения, а также измерить ток и напряжения потребителя;
* CFluke – класс для управления мультиметром FLUKE 8846A. Позволяет соединиться с мультиметром, измерить сопротивление и электропроводность;
* CAkip – класс для управления генератором сигналов АКИП 3402. Позволяет соединиться, включить, задать и измерить частоту сигнала;
* CMensor – класс для управления калибратором давления Mensor CPC600. Позволяет соединиться, включить, подать и получить значение давления, а также получить значение стабильности давления (0 или 1).

Диаграмма классов программы представлена на рисунке 1.6.

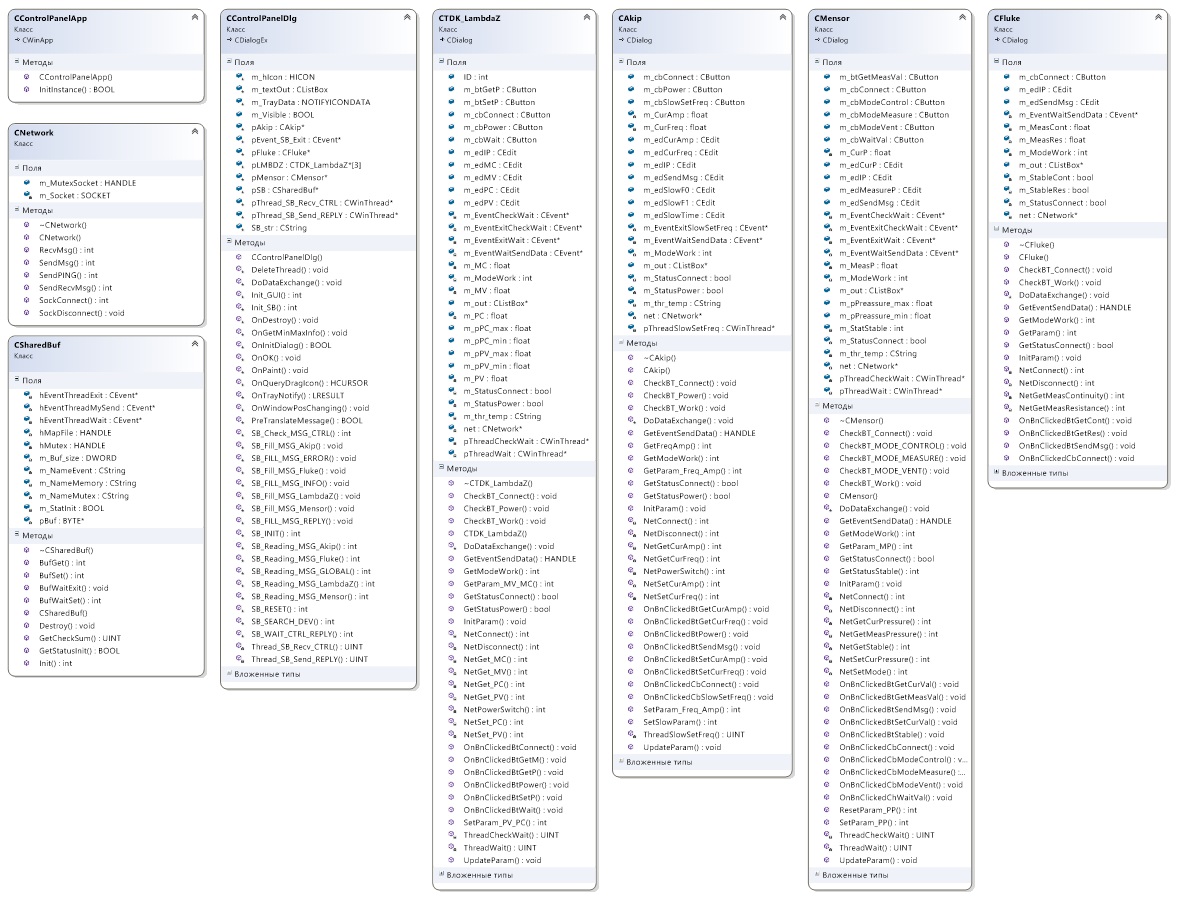


Рис. 1.7 - Диаграмма классов программы DeviceManagment

**1.3.3. Описание алгоритма работы тестовых программ**

Программы-тесты являются отдельными самостоятельными исполняемыми файлами, которые запускает программа-менеджер CheckFut. Тесты запускаются с соответствующими параметрами командной строки, взятыми из config.dat, в автоматическом режиме. Однако, если запустить тест не из-под программы-менеджера, а стандартными средствами Windows, то он будет запущен в ручном режиме с расширенными возможностями настройки и передачи информации. Различия в интерфейсах показаны на рисунках 1.8 и 1.9.

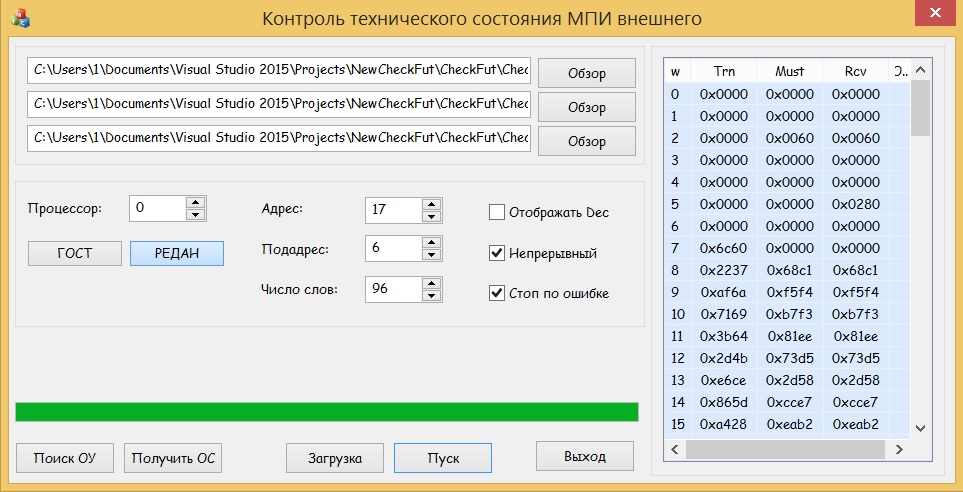


Рис. 1.8 - Ручной режим работы программы MkioTest

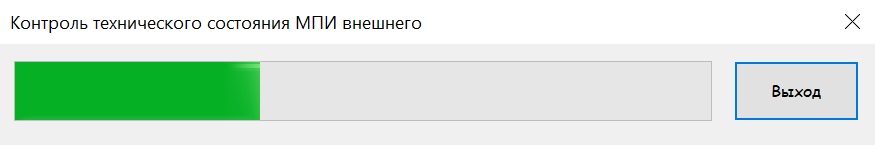


Рис. 1.9 - Автоматический режим работы программы MkioTest

В зависимости от выбранного в головной программе типа изделия (модуль аппаратурный, блок электронный или модуль контроллеров управления) и типа хранения (в контейнере или без) некоторые тесты могут оказаться недоступны. Например, если изделие находится в контейнере, то нет возможности подключить к нему калибратор давления, так как заборники находятся с внешней стороны подводного аппарата. Соответственно нет никакого смысла проверять работоспособность датчиков давления. Всего же было написано 19 тестов:

* MkioTest – программа контроля МПИ внешнего, то есть интерфейса, соединяющего стойку с изделием через адаптер TA1-USB-01-М;
* TypeTest – программа контроля команд и признаков. Программа не задействует приборы, а лишь посылает команды на оконечное устройство в изделии, проверяя необходимые признаки;
* VoltageTest – программа контроля напряжений. В изделии несколько линий тока, на каждой из которых должно быть определенное напряжение. С помощью мультиметра этот тест проверяет их все;
* HiVoltageTest – программа контроля высокого напряжения. Программа проверяет корректность выдаваемого высокого напряжения в 200 В, выставляя на изделии уставку с определенным коэффициентом;
* LinkTest – программа проверки канало Link между процессорами. Так как на платах имеется несколько процессоров, взаимодействующих друг с другом, то необходимо проверить межпроцессорный канал. Тест никак не задействует стойку, а только посылает массивы данных с одного процессора на другой, проверяя его достоверность;
* FramTest – программа проверки энергонезависимой памяти FRAM при каждом процессоре платы МКУ (всего на ней 3 процессора). Тест, который не задействует стойку, а записывает данные в память и проверяет их.
* SpeedTest – программа контроля оборотов изделия, имитирующая работу двигателя. С генератора посылается сигнал определенной частоты и амплитуды, а программа проверяет насколько корректно он воспринимается на изделии.
* DadTest – программа контроля работоспособности датчиков давления. Калибратор давления поочередно подает 3 заранее определенных величины давления на изделие. Программа же, будучи запущенной 3 раза, проверяет уложилось ли это значение в допуски.
* BchRifTest – программа контроля работоспособности блока чувствительных элементов. При включении БЧЭ контролируется повышение тока потребления, тест же контролирует точность гироскопов и акселерометров в допустимых пределах;
* NiisTest – программа контроля одного из магнитометров. Так как в составе проверяемого изделия имеется 2 магнитометра, каждый со своей частотой обмена, чувствительностью и протоколом взаимодействия, то было решено реализовать по тесту на каждый из них. Программа не взаимодействует со стойкой, а лишь снимает показания с осей магнитометра, высчитывает модуль напряженности и сравнивает его с допустимыми пределами отклонения.
* RadarTest – программа контроля второго магнитометра (трехкомпонентного преобразователя). Принцип работы такой же, как и у предыдущей программы, отличия лишь в протоколе обмена.
* RsStubTest – программа контроля информационного канала с блоком управления кормовой. Так как, настоящий блок не используется при контроле работоспособности, то необходимо проверить хотя бы канал RS-485. Для этого на выход под блок ставится специальная заглушка, замкнутая сама на себя. Тест посылает контрольный массив и принимает его же, проверяя достоверность.
* RegTest – программа контроля работоспособности регистраторов МКУ. При испытаниях и реальной эксплуатации изделие записывает регистрацию, которая в дальнейшем анализируется на предмет наличия ошибок ходовых алгоритмов. Эта регистрация записывает во внутреннее ПЗУ, которое необходимо проверить. Тест записывает, считывает, проверяет и стирает данные на регистраторах, проверяя их на битые блоки. Если количество таких блоков оказывается выше определенного значения, то изделие считается неработоспособным;
* TuBridgeTest – программа контроля технического состояния канала телеуправления. Так как изделие может передвигаться не только автономно, но и под управлением оператора, то необходимо проверить канал, по которому аппарат будет управляться. Тест имитирует различные команды телеуправления, преобразуя их в коды АЦП и проверяя не вышли ли они за допустимые пределы;
* MkioIsaTest – программа контроля работоспособности МПИ внутреннего. В изделии имеются 2 модуля ТАМ1-ISA от компании «Элкус», предназначенных для работы с резервированным мультиплексным каналом. По такому каналу подключается одно из оконечных устройств к изделию. Так как настоящее оконечное устройство не используется при проверке работоспособности, то один модуль используется в качестве контроллера канала, а второй – в качестве оконечного устройства. Также для полноты проверки следует поменять их местами, поэтому тест запускается 2 раза. Программа посылает данные по каналу и проверяет их достоверность по приходу на ОУ;
* RsChannelTest – программа контроля канала информационного обмена. При проверке БЭ реальные датчики не используются: вместо них берутся адаптеры RS-485, которые соединяются с ПК по интерфейсу USB. Программа посылает массив данных через канал и принимает через адаптер на ПК, проверяя их целостность и достоверность. Тест запускается 5 раз для двух датчиков давления, двух магнетометров и БЧЭ;
* Test\_FmMvu – программа контроля работоспособности регистраторов платы модуля внутренних устройств (МВУ). Плата МВУ используется для обработки гидроакустики и управления движением подводного аппарата. На ней также имеются свои процессоры, регистраторы, ОЗУ и ППЗУ. Проверка регистраторов похожа на аналогичную на плате МКУ: точно так же программа записывает, считывает, проверят и очищает данные проверяя аппаратуру на битые блоки;
* Test\_RamMvu – программа контроля ОЗУ МВУ. Программа проверяет оперативную память по схожей схема, что и регистраторы, сначала записывает данные, потом считывает, проверяя целостность и достоверность. Однако проверка проходит значительно быстрее;
* Test\_RomMvu – программа контроля ППЗУ МВУ. В эту память в дальнейшем будет записано бортовое программное обеспечение (БПО), поэтому этот тест особенно важен. Проверка аналогична проверке регистраторов за исключением тестовой записи бортовой программы и обмена с ней. После успешного обмена, ППЗУ очищается, а изделие считается работоспособным.

Каждый из перечисленных тестов имеет одну или несколько технологических программ, загружаемых во внутреннюю память процессоров российского производства «Миландр» 1967ВН028. Эти программы написаны в среде программирования Visual DSP 5.0 и скомпилированы под ближайший аналог – процессор ADSP-TS201. При автоматическом запуске теста технологические программы загружаются на процессор, производится контрольный обмен, далее обмен данными согласно алгоритмам, проверка на допустимость значений и формирование протокола. Технологические программы написаны на чистом языке С из-за ограничений, накладываемых процессором.

**1.3.4. Описание библиотеки формирования протокола**

По окончании работы каждая тестовая программа формирует протокол в формате .txt. Программа-менеджер CheckFut после окончания последнего теста объединяет все эти текстовые протоколы и записывает в формате .pdf с защитой от изменения и копирования. За этот функционал отвечает библиотека libhpdf.dll – надстройка над публичной кроссплатформенной библиотекой LibHaru, обладающая дополнительным функционалом. Оригинальная библиотека предоставляет множество функций для работы с pdf-файлами таких как настройка ограничений, задание параметров шрифта, средства для рисования, поэтому выбор пал именно на нее. Однако ни в одной подобной библиотеке не было средств для работы с таблицами, из-за чего было принято решение написать надстройку и скомпилировать ее в удобный формат .dll для подключения к любой программе. Дополнительные функции были реализованы в классе HaruClass, который должен включаться в проект. Именно через него программист инициализирует библиотеку, добавляет текст, строит таблицы, редактирует колонтитулы и прочее. Структура класса представлена в общей диаграмме классов программы CheckFut. Пример протокола приведен в приложении.

# 1.4. Организация данных

# 1.4.1. Организация входных данных

Входными данными для программ-тестов являются:

* ini-файлы конфигурации;
* технологические программы для загрузки на процессор.

Входными данными для программы работы со стойкой DeviceManagement являются:

* plink.exe – интерфейс командной строки для клиента удаленного подключения к серверам PuTTY;
* ini-файлы с IP и MAC-адресами устройств;
* config.ini – файл конфигурации с начальными значениями и ограничениями на приборы.

Входными данными для программы-менеджера CheckFut являются:

* файл настроек тестов config.dat;
* исполняемые файлы программ-тестов;
* библиотека для создания протокола libhpdf.dll;
* коды при нажатии клавиш оператором.

# 1.4.2. Организация выходных данных

Выходными данными для программ-тестов являются:

* результаты контроля работоспособности, выведенные на экран, и в бинарном виде;
* текстовые файлы отчета с метками для объединения.

Выходными данными для программы работы со стойкой DeviceManagement является запись структуры в общую память о состоянии приборов согласно протоколу взаимодействия.

Выходными данными для программы-менеджера CheckFut являются:

* общий результат контроля работоспособности;
* диагностические окна об ошибках;
* защищенный файл отчета в формате .pdf.

# 1.5. Требования к составу и параметрам технических средств

Для функционирования программы необходимо наличие следующих технических средств, обусловленное минимальной поддерживаемой ОС MS Windows Vista:

* процессор Intel Pentium 4 (с тактовой частотой не менее 1.0 ГГц) или выше;
* ОЗУ объемом не менее 512 Мб;
* видеокарта с поддержкой DirectX 9 и разрешения экрана не менее 1024х768;
* свободное пространство на жестком диске не менее 0.2 Гб;
* наличие свободного интерфейса Ethernet;
* манипулятор типа “мышь” и клавиатура.

# 1.6. Требования к информационной и программной совместимости

Для эксплуатации программы ЭВМ должна работать под операционной системой MS Windows Vista/7/8/8.1/10 и новее. Для разработки программного продукта необходима интегрированная среда разработки от корпорации Microsoft – MS Visual Studio Ultimate.

# 2. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ И ИХ ОЦЕНКА

# 2.1. Объект испытаний

Объектом испытаний является программа контроля технического состояния CheckFut, назначением которой является запуск программ-тестов в соответствии с выбранным режимом проверки, обмен данными с программой управления приборами DeviceManagment и формирование защищенного файла-протокола по результатам контроля.

# 2.2. Цель испытаний

Цель проведения испытаний – проверка соответствия характеристик разработанной программы функциональным требованиям, предъявляемым к программно-аппаратному комплексу.

# 2.3. Требования к программе

Программа должна выполнять следующие функции:

* проверка подключения стойки с приборами;
* обеспечение выбора режима работы (место проведения, тип изделия, тип хранения);
* управление измерительными приборами стойки через программу DeviceManagement;
* запуск тестовых программ в соответствии с выбранным режимом;
* обмен данными с тестовыми программами для получения результата проверок и отображения прогресса контроля;
* обработка аварийных ситуаций в процессе работы и выдача диагностических сообщений;
* формирование защищенного протокола по результатам работы;
* отображение протоколов прошлых запусков программы.

# 2.4. Средства и порядок проведения испытаний

# 2.4.1. Технические средства, используемые во время испытаний

Для проведения испытаний использовался ПК, обладающий следующими техническими параметрами:

* процессор Intel Core i5 с тактовой частотой 2.4 ГГц;
* ОЗУ объемом 6 Гб;
* видеокарта Intel HD Graphics;
* экран с разрешением 1920х1080;
* манипулятор типа «мышь» и клавиатура.

Также в состав технических средств входила стойка со следующими измерительными приборами:

* преобразователь Advantech EKI-1528;
* коммутатор LAN MOXA PowerTrans PT-7728;
* четыре источника постоянного тока TDK LAMBDA GEN-40-19;
* генератор сигналов АКИП 3402;
* кондиционер сетевого питания Furman PL-8C-E;
* многофункциональный блок Agilent 34980A;
* мультиметр FLUKE 8846A;
* калибратор давления Mensor CPC600.

# 2.5.2. Программные средства, используемые во время испытаний

Для проведения испытаний использовался ПК, работающий под операционной системой MS Windows 10.

# 2.6. Методы испытаний

Тестирование проводилось стратегией «чёрного ящика». В этом методе программа рассматривается как объект, внутренняя структура которого неизвестна. Тестировщик вводит данные и анализирует результат, но, как именно работает программа, он не знает[6]. Результаты тестирования приведены в таблице 2.1.

Таблица 2.1 - Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение | Входные данные | Эталон | Результат |
| Проверка нахождения файла конфигурации config.dat в папке с программой | Файл находится в папке | Окно программы не отображается. Вывод диагностического окна: «Не найден файл конфигурации» | Совпадает |
| Файл не находится в папке | Запуск программы | Совпадает |
| Проверка введенных текстовых данных | Ввод числовых значений и знаков препинания в текстовые поля | Невозможно ввести знак | Совпадает |
| Ввод текста в поле «Номер изделия» | Невозможно ввести знак | Совпадает |
| Ввод текста меньше 2 знаков | Кнопка «Далее» заблокирована | Совпадает |
| Корректный ввод в текстовые поля | Активация кнопки «Далее» | Совпадает |
| Выбор режима работы | МА | Возможно выбрать тип хранения. Активация кнопки «Далее» | Совпадает |
| БЭ | Невозможно выбрать тип хранения. Активация кнопки «Далее» | Совпадает |
| МКУ | Невозможно выбрать тип хранения. Активация кнопки «Далее» | Совпадает |
| Проверка окна «Архив протоколов» | Ввод типа и номера изделия | Вывод списка протоколов согласно параметрам | Совпадает |
| Ввод даты начала и конца | Вывод стандартного окна выбора даты. Вывод списка протоколов согласно параметрам | Совпадает |
| Нажатие кнопки «Просмотр» | Открытие протокола в программе просмотра pdf-файлов по умолчанию | Совпадает |

Продолжение таблицы 2.1 - Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение | Входные данные | Эталон | Результат |
| Проверка окна «Настройки» | Изменение содержимого в ячейке | Сохранение новой информации в файле config.dat | Совпадает |
| Проверка подключения | Стойка не подключена к ПК. Нажатие на кнопку «Начать контроль» | Вывод диагностического сообщения: «Не найдены приборы для проверки» | Совпадает |
| Стойка подключена к ПК. Отсутствуют один или несколько приборов | Вывод диагностического сообщения: «Не найдены следующие приборы: … Продолжить контроль?». Если да, то далее работа в автоматизированном режиме | Совпадает |
| Стойка подключена к ПК. Нажатие на кнопку «Начать контроль» | Работа в автоматическом режиме | Совпадает |
| Проверка исполняемых файлов файлов-тестов | Файлы не находятся в папке с программой | Вывод диагностического сообщения: «Не найдены исполняющие файлы для следующих программ: … Перейти в настройки для выбора файлов?» | Совпадает |
| Файлы находятся в папке с программой | Работа в автоматическом или автоматизированном режиме | Совпадает |
| Проверка автоматического режима | Материальная часть полностью работоспособна | Выдача тестами сообщений «Работоспособен» по результатам проверок. Выдача сообщения «Техническое состояние работоспособное» по окончании | Совпадает |

Продолжение таблицы 2.1 - Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение | Входные данные | Эталон | Результат |
| Проверка автоматизирован-ного режима | Одна или несколько подсистем неработоспособны | Выдача соответствующих тестам сообщений «Неработоспособен» по результатам проверок. Выдача сообщения «Техническое состояние неработоспособное» по окончании | Совпадает |
| Нажатие на кнопку «Начать контроль» | Появление окна контроля напряжения и тока. Ввод и контроль значений | Совпадает |
| Проверка автоматизирован-ного режима | Запуск теста «Напряжения» | Последовательное появление окон «Контроль напряжения». Ввод и контроль значений | Совпадает |
| Запуск теста контроля оборотов | Появление окна «Установите режим работы генератора» | Совпадает |
| Окончание теста контроля оборотов | Появление окна «Отключите генератор» | Совпадает |
| Запуск теста «Датчики давления» | Последовательное появление окон «Установите давление» | Совпадает |
| Проверка окна контроля напряжения и тока | Ввод букв и знаков препинания, кроме «.» | Невозможно ввести знак | Совпадает |
| Ввод некорректных значений напряжения или тока | Вывод диагностического окна «Несоответствие заданному диапазону» | Совпадает |
| Ввод корректных значений напряжения и тока | Запуск тестов | Совпадает |

Продолжение таблицы 2.1 - Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение | Входные данные | Эталон | Результат |
| Проверка окна контроля напряжений в тесте напряжений | Ввод некорректного значения напряжения | Вывод диагностического окна «Несоответствие заданному диапазону» | Совпадает |
| Ввод корректного значения напряжения | Продолжение контроля | Совпадает |
| Проверка протокола | Файл libhpdf.dll не находится в папке с программой | Вывод диагностического сообщения «Ошибка сохранения файла протокола» по окончании проверок | Совпадает |
| Файл libhpdf.dll находится в папке с программой. Нажатие кнопки «Протокол» по окончании контроля | Запуск программы по умолчанию для открытия pdf-файла | Совпадает |
| Проверка протокола | Нажатие кнопки «Прекратить контроль» до окончания проверки | Файл протокола сформирован только из пройденных тестов | Совпадает |
| Удаление файлов протоколов в процессе проверки | Вывод диагностического сообщения «Не найден файл протокола … теста … для занесения в общий протокол» по окончании проверок | Совпадает |
| Проверка доступности тестов при разных типах и комплектациях | Изделие МА | Недоступны тесты контроля канала информационного обмена | Совпадает |
| Изделие БЭ | Недоступны тесты контроля ДД, БЧЭ, магнитометров и БУКа | Совпадает |

Продолжение таблицы 2.1 - Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Назначение | Входные данные | Эталон | Результат |
| Проверка доступности тестов при разных типах и комплектациях | Изделие МКУ | Недоступны тесты контроля ДД, БЧЭ, магнитометров и БУКа, канала информационного обмена, плат КС и ДЗ | Совпадает |
| Изделие МА в контейнере | Недоступны тесты контроля канала информационного обмена, ДД, регистраторов, плат КС и ДЗ | Совпадает |
| Проверка меню «О программе» | Выбор в меню  «О программе» | Вывод диалогового окна с информацией о программе | Совпадает |
| Проверка кнопки «Выход» | Нажатие кнопки выход | Закрытие программы | Совпадает |

В тестовых примерах, приведенных в таблице 2.1, фактическая и ожидаемая реакции программы совпадают, что подтверждает корректность и работоспособность программы. Заявленные в техническом задании требования к программе полностью выполнены.

# 3. ОПИСАНИЕ ОКОН И ЭЛЕМЕНТОВ ПРОГРАММЫ

# 3.1. Описание главного окна программы

Главное окно программы состоит из нескольких вкладок, переключаться между которыми можно с помощью кнопок «Назад» и «Далее». Вид начальной вкладки «Ввод служебной информации» представлен на рисунке 3.1.

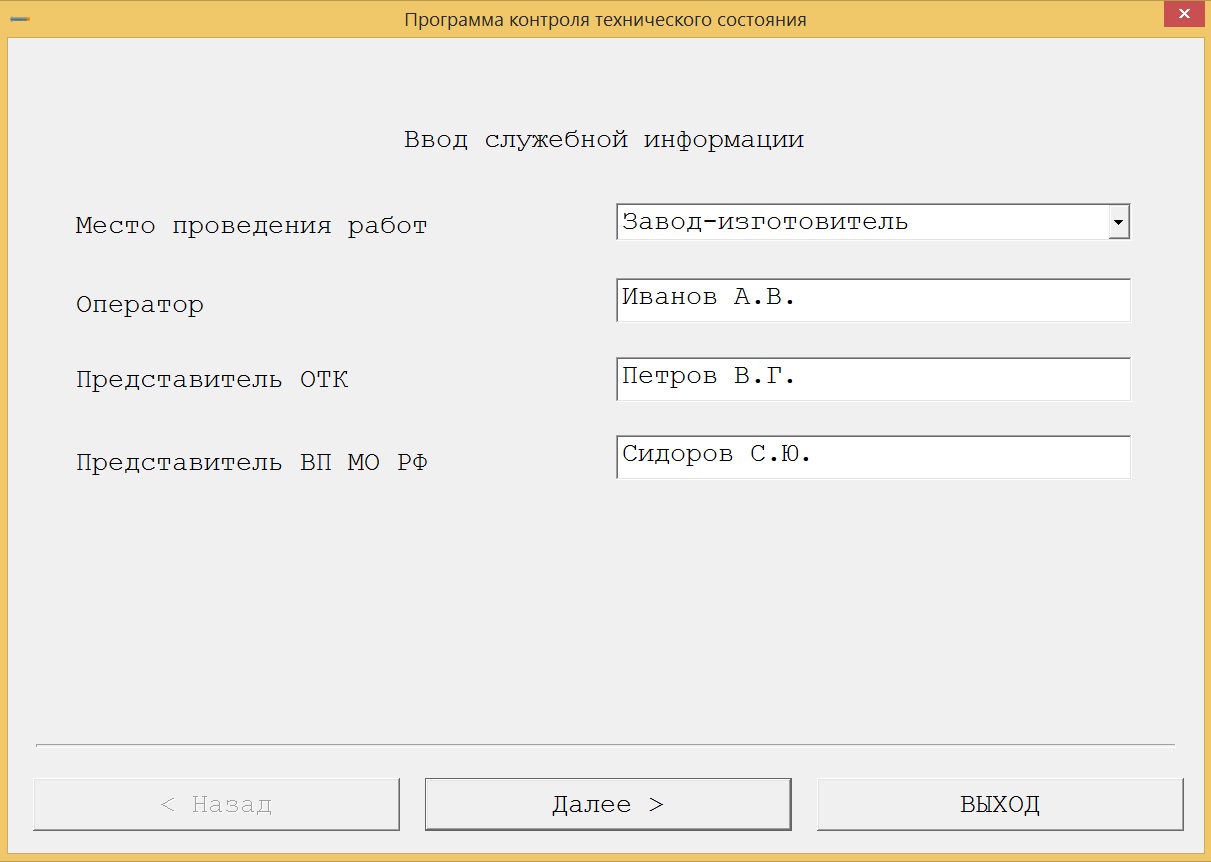


Рис. 3.1 - Вкладка «Ввод служебной информации»

Выпадающий список «Место проведения работ» предназначен для выбора между заводом-изготовителем и технической позицией флота. От этого зависит вид нескольких окон. Текстовые поля «Оператор», «Представитель ОТК», «Представитель ВП МО РФ» предназначены для ввода соответствующей информации для дальнейшего занесения в протокол. Кнопка «Выход» на каждой вкладке закрывает программу.

После ввода всех данных и нажатия кнопки «Далее» открывается вкладка «Выбор режима работы», представленная на рисунке 3.2.

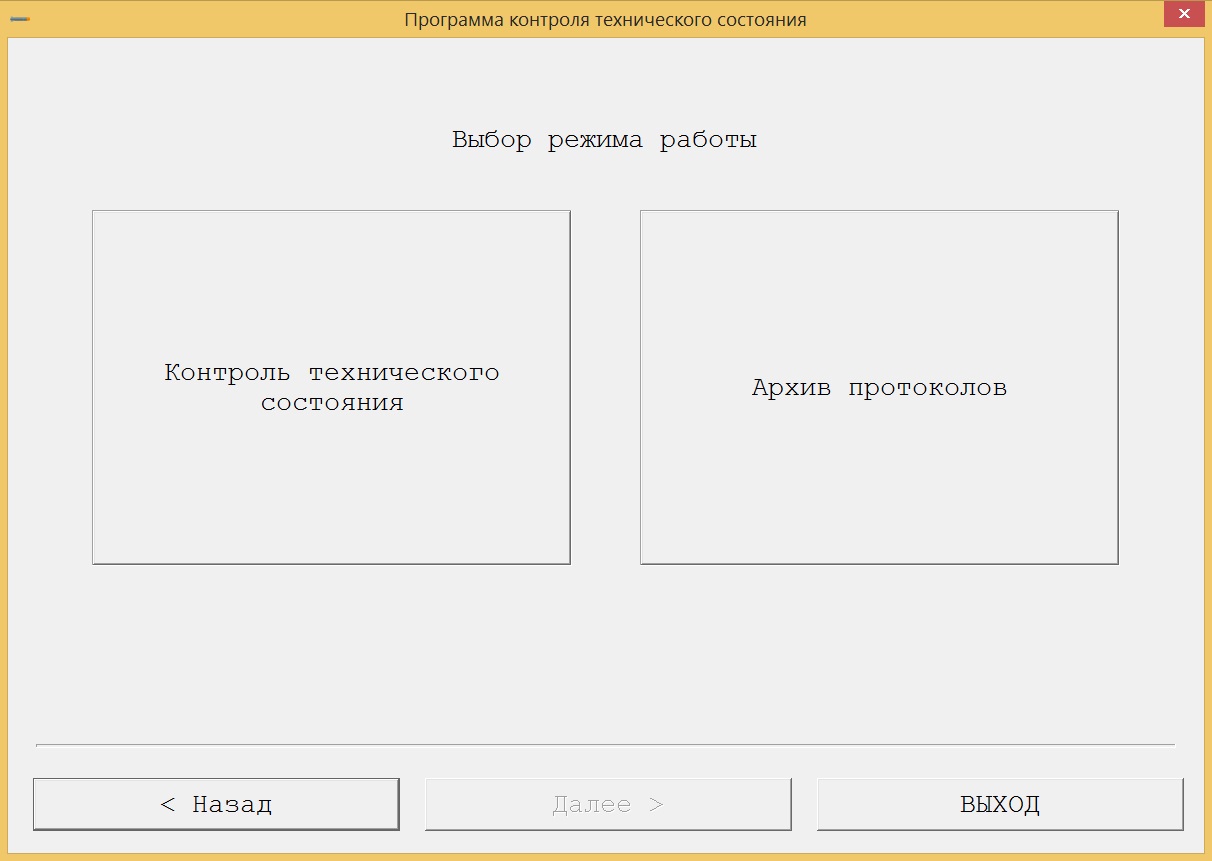


Рис. 3.2 - Вкладка «Выбор режима работы»

Данная вкладка предоставляет выбор между дальнейшим продолжением проверки и просмотром архивов протоколов. После выбора активируется кнопка «Далее».

Вкладка «Архив протоколов» представлена на рисунке 3.3. Она позволяет просмотреть любой из протоколов прошлых проверок стандартной программой просмотра pdf-файлов. Для этого необходимо два раза кликнуть по интересующему протоколу или выбрать его и нажать кнопку «Просмотр». Текстовое поле «Тип изделия» предназначено для сортировки отчетов только по типу, введенному в данное поле. Текстовое поле «Номер изделия» предоставляет ту же функцию, но только по номеру. Текстовые поля «Дата (дд.мм.гггг)» позволяет вывести в список отчеты только за указанные даты: от и до. Кнопки «…» рядом с этими текстовыми полями вызывают стандартное окно выбора даты, изображенное на рисунке 3.4.

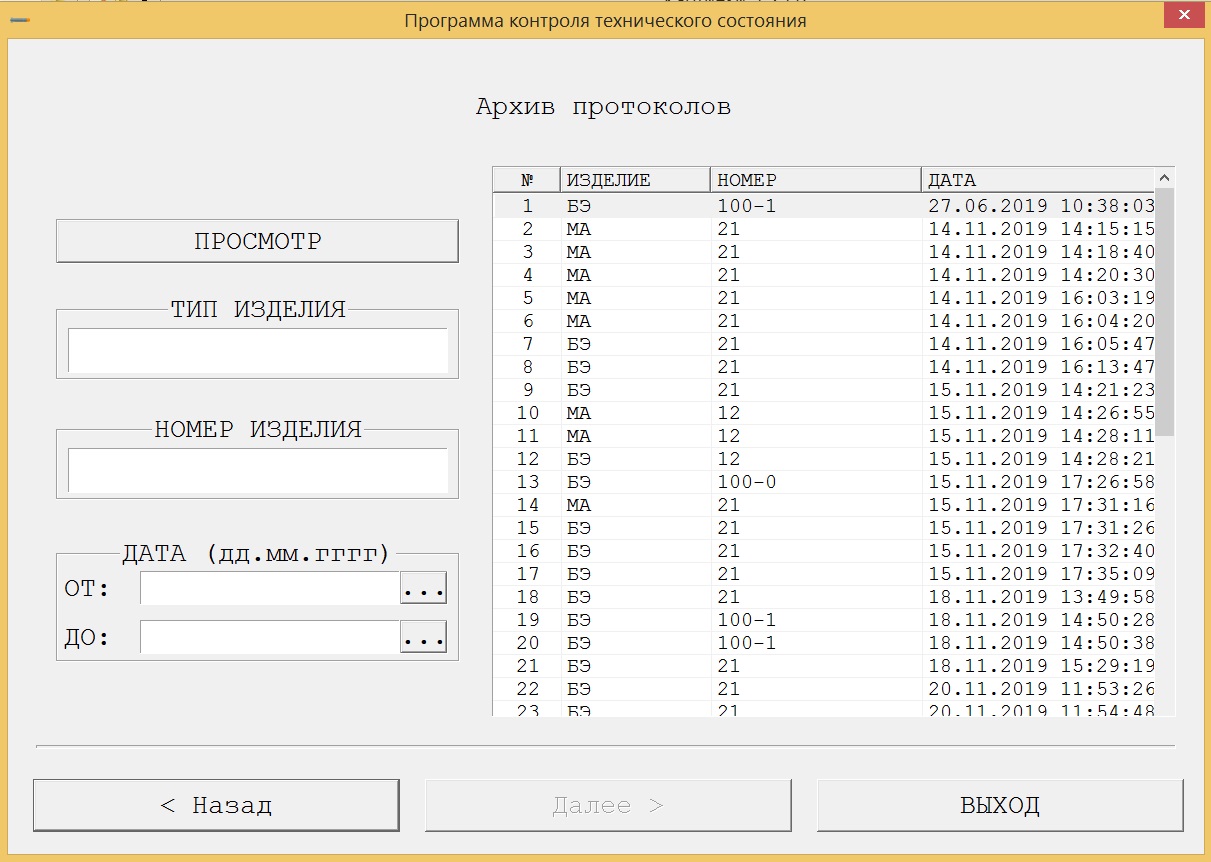


Рис. 3.3 - Вкладка «Архив протоколов»

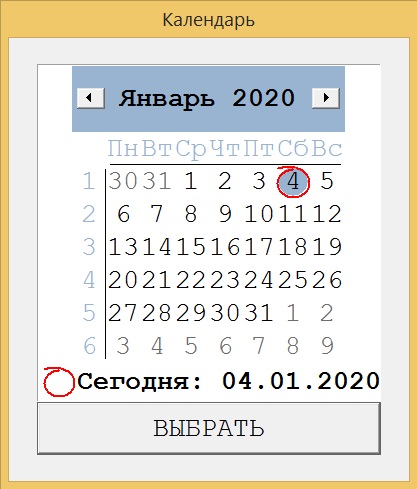


Рис. 3.4 - Окно выбора даты

Дальнейшим продолжением проверки является вкладка «Ввод данных объекта контроля», изображенная на рисунке 3.5.

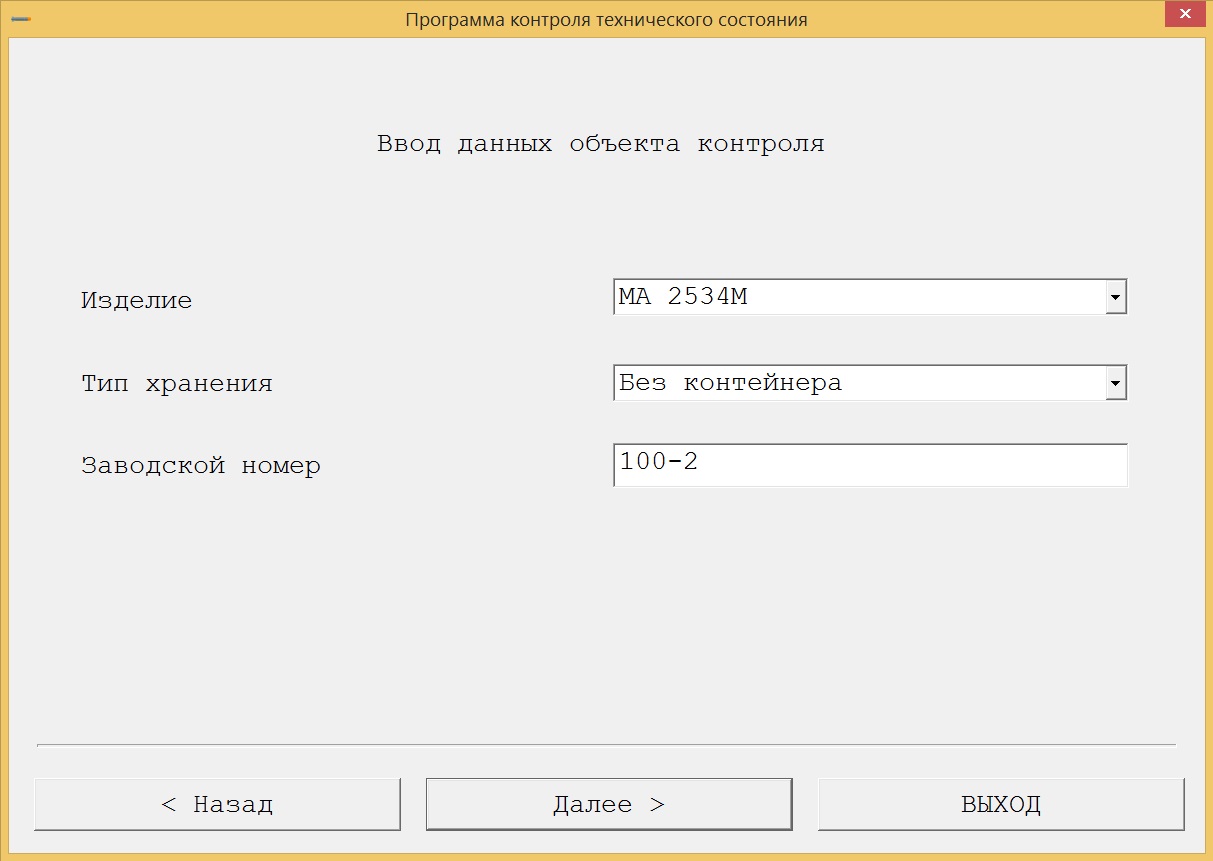


Рис. 3.5 - Вкладка «Ввод данных объекта контроля»

Выпадающий список «Изделие» предоставляет выбор материальной части, которую необходимо проверить МА, БЭ или МКУ. Выпадающий список «Тип хранения» отвечает за выбор комплектации изделия: в контейнере или без контейнера. Этот выбор предоставляется только для изделия МА, для остальных изделий данный список заблокирован, как показано на рисунке 3.6. Текстовое поле «Заводской номер» предназначено для ввода номера изделия, который в дальнейшем будет занесен в протокол.

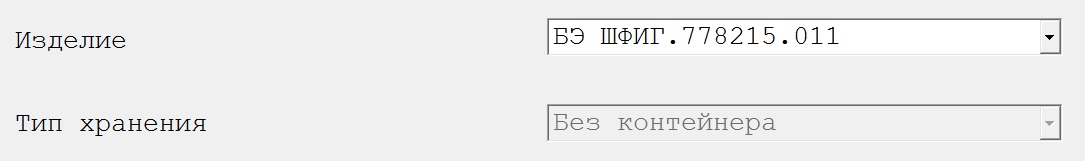


Рис. 3.6 - Блокировка списка «Тип хранения»

После ввода всех данных и нажатия кнопки «Далее» открывается вкладка «Контроль технического состояния», представленная на рисунке 3.7.

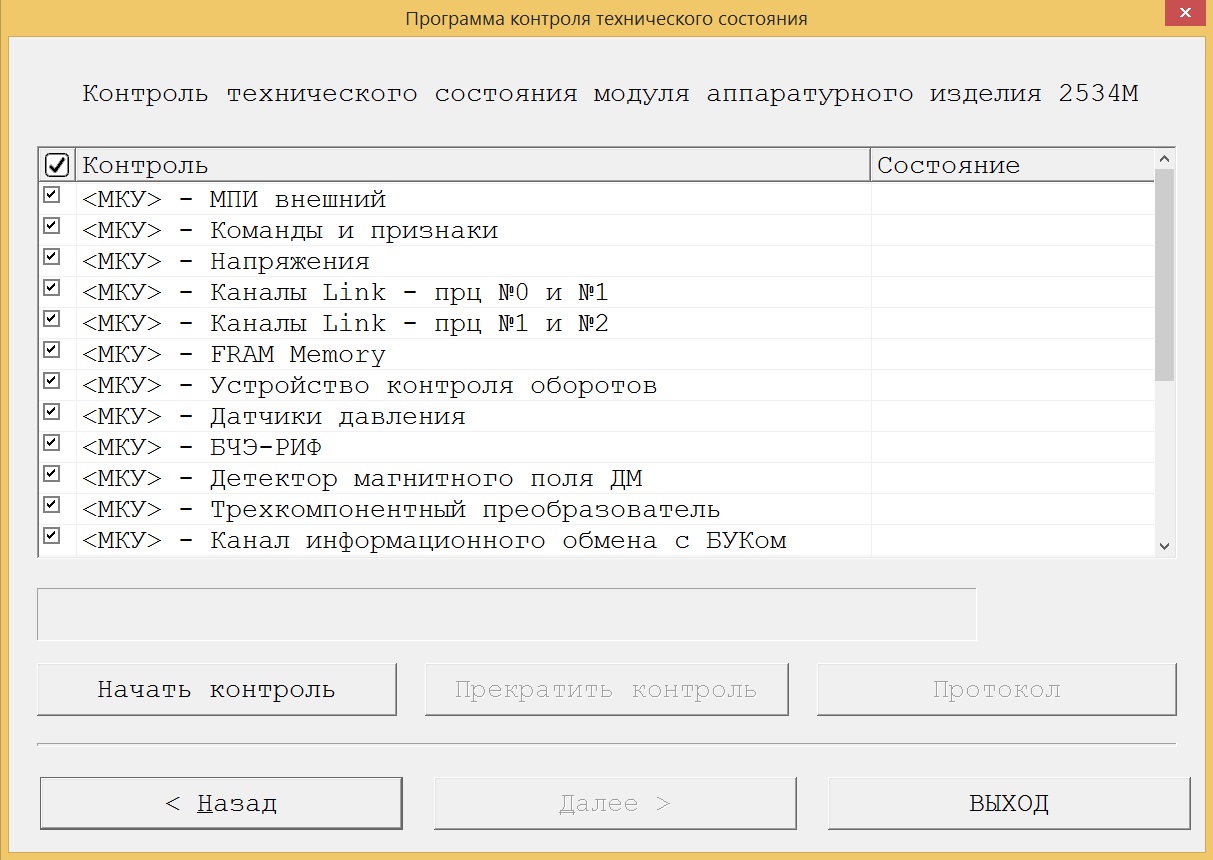


Рис. 3.7 - Вкладка «Контроль технического состояния»

Флажки ☑ позволяют убрать или выбрать определенные тесты. Кнопка «Начать контроль» инициирует проверку подключенных устройств и запуск тестов. Кнопка «Прекратить контроль» становится активной после начала проверки. При ее нажатии происходит экстренное прекращение всех проверок. Кнопка «Протокол» становится активной после окончания всех проверок. При ее нажатии открывается протокол по результатам контроля программой по умолчанию для pdf-файлов. Пример протокола приведен в приложении.

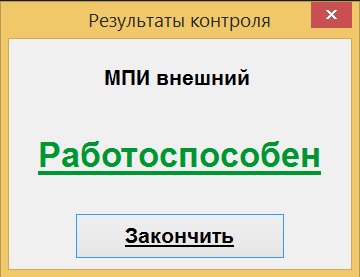
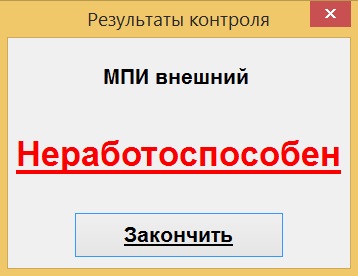
# 3.2. Описание вспомогательных окон программы

После нажатия на кнопку «Начать контроль» проверки выполняются в автоматическом режиме. Тесты поочередно запускаются, процесс выполнения отображается в окне, показанном на рисунке 3.8.



Рис. 3.8 - Вид окна «Контроль технического состояния МПИ внешнего»

Результаты тестов отображаются с помощью окна, показанного на рисунке 3.9 если подсистема работоспособна, или окна, показанного на рисунке 3.10 если подсистема неработоспособна.



|  |  |  |
| --- | --- | --- |
| Рисунок 3.9 - Вид окна «Результаты контроля» |  | Рисунок 3.10 - Вид окна «Результаты контроля» |

Окна, аналогичные данным, появляются для каждого теста контроля, за исключением тестов «Датчики давления» и «БЧЭ-РИФ», и отображает ход выполнения контроля. Над индикатором процесса выводится соответствующее наименование выполняемого теста. Каждый тест заканчивается появлением окна с соответствующим результатом выполненного теста: «Работоспособен» или «Неработоспособен».

В процессе окончания каждого теста контролировать засвечивание строки зеленым цветом и появление надписи: «РАБОТОСПОСОБНОЕ» в колонке «Состояние» (рисунок 3.11) или контролировать засвечивание строки красным цветом и появлением надписи: «НЕРАБОТОСПОСОБНОЕ» (рисунок 3.12).

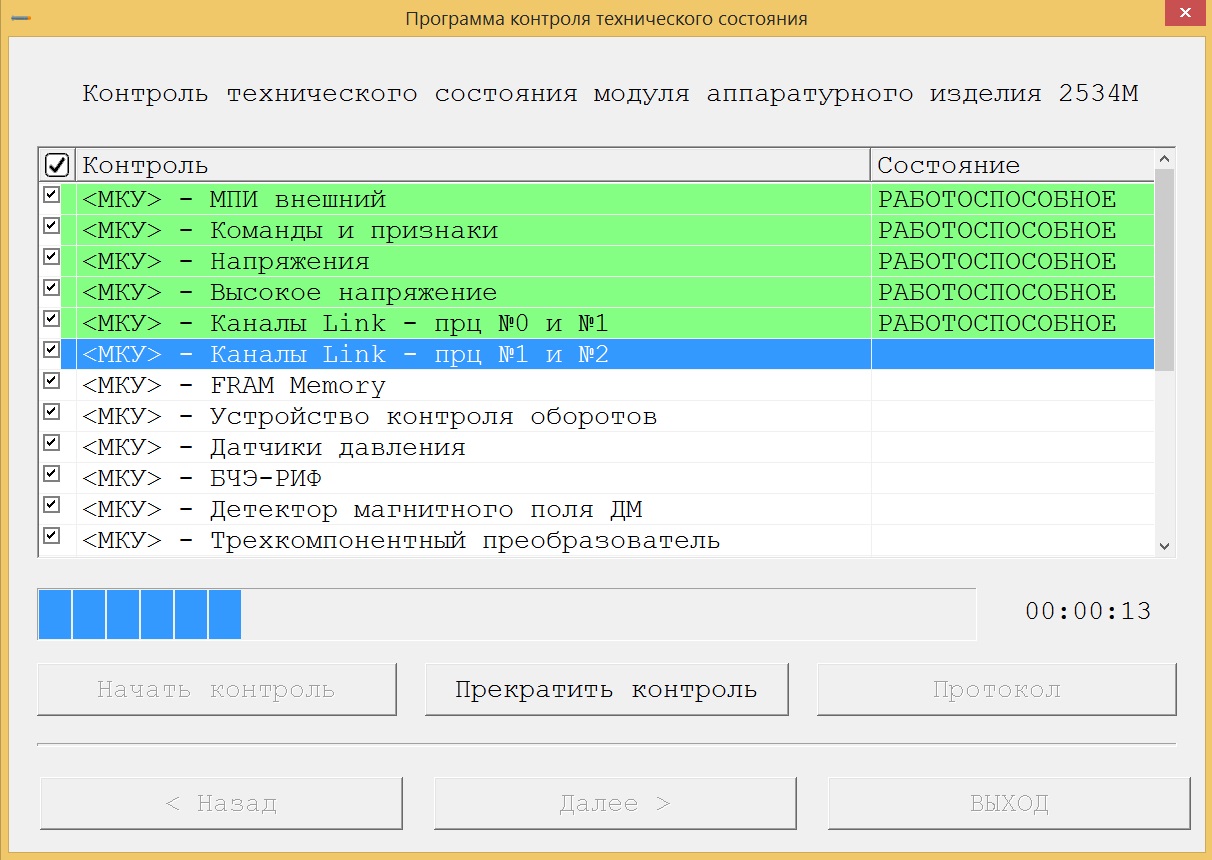


Рис. 3.11 - Вкладка «Контроль технического состояния». Состояние работоспособное

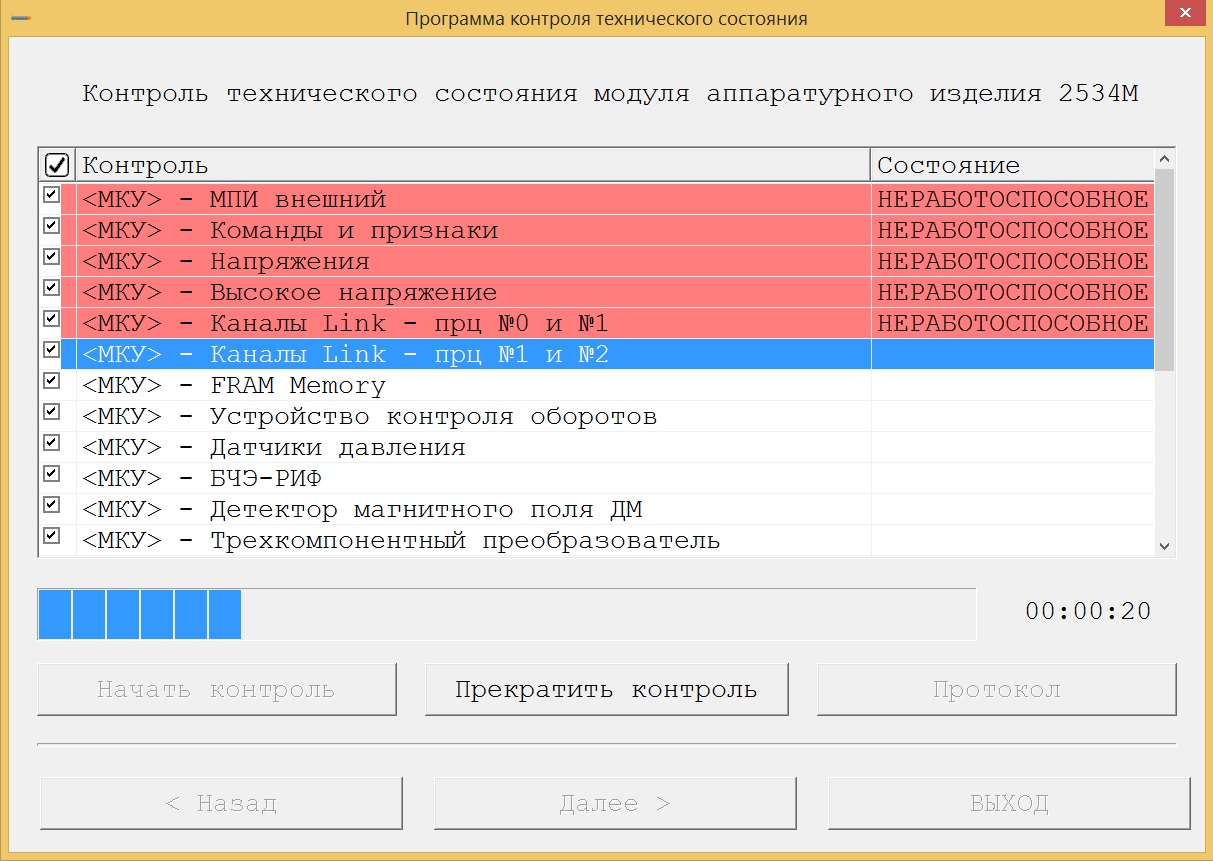


Рис. 3.12 - Вкладка «Контроль технического состояния». Состояние неработоспособное

По окончании процедуры контроля на экране ПК появляется окно, показанное на рисунке 3.13, если техническое состояние работоспособное, в противном случае – окно, показанное на рисунке 3.14.

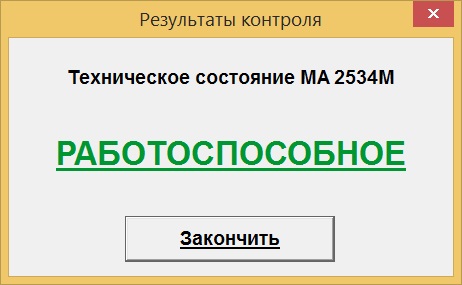


Рисунок 3.13 - Вид окна «Результат контроля». Состояние работоспособное



Рисунок 3.14 - Вид окна «Результат контроля». Состояние неработоспособное

Если же после нажатия на кнопку «Начать контроль» от одного или нескольких приборов нет ответа, то работа продолжается в автоматизированном режиме. Например, если не обнаружен источник питания, то необходимо будет включить его вручную. В таком случае на экране будет выведено окно, изображенное на рисунке 3.15. Оператору необходимо самому ввести значения напряжения и тока на источнике, а программа проверит их на корректность.

При выполнении теста «Напряжения» на экране ПК будут появляется окна, аналогичные окну, показанному на рисунке 3.16 для разных напряжений. Оператору необходимо ввести измеренное напряжение.

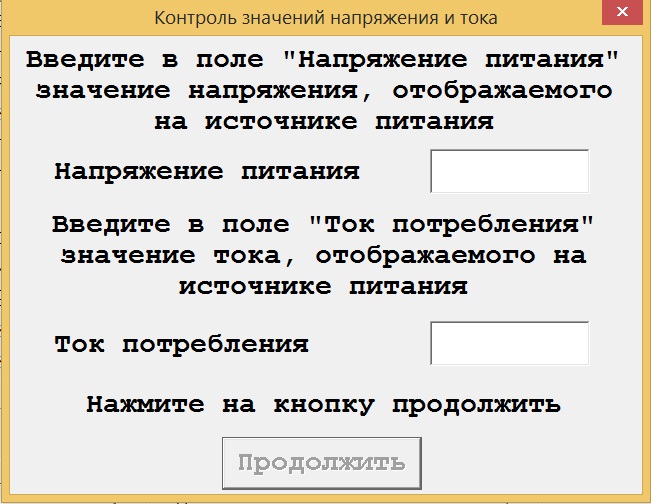


Рис. 3.15 - Окно контроля значений напряжения и тока

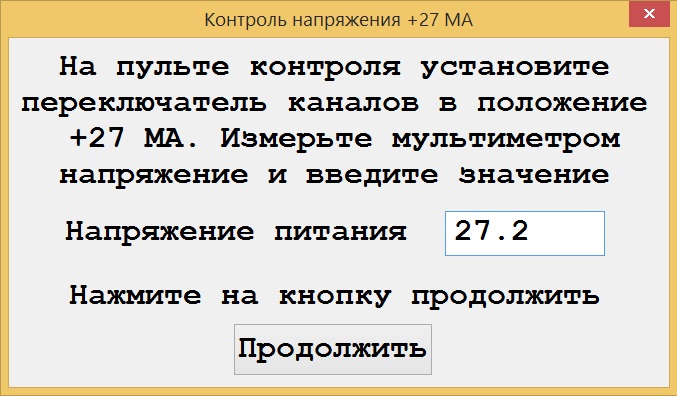


Рисунок 3.16 - Окно контроля напряжения +27 МА

В случае несоответствия введенного значения требуемому диапазону экране ПК появится окно, показанное на рисунке 3.17. Такое окно будет появляться при каждом несоответствии введенного значения требуемому диапазону для каждого параметра в объеме теста «Напряжения».

При выполнении теста «Устройство контроля оборотов» на экране ПК появится окно, показанное на рисунке 3.18.

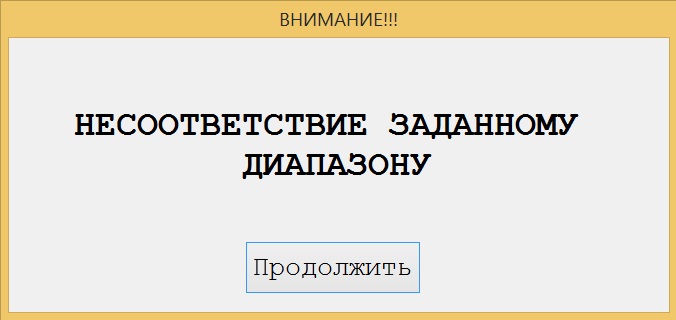


Рисунок 3.17 - Окно «Несоответствие заданному диапазону»

Оператору необходимо вручную выставить требуемые параметры генератора. После завершения теста на экране ПК появится окно, изображенное на рисунке 3.19. Оператору необходимо выключить генератор.

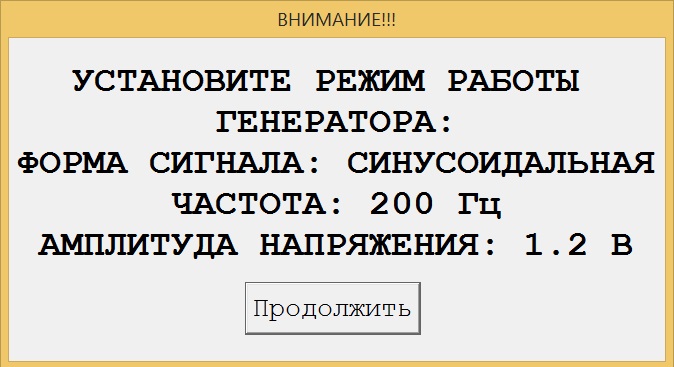


Рисунок 3.18 - Окно «Установите режим работы генератора»

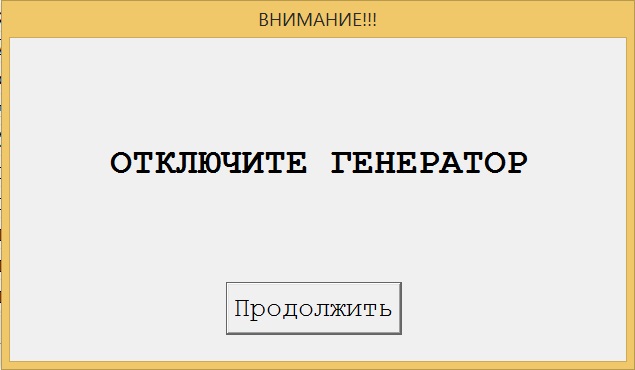


Рисунок 3.19 - Окно «Отключите генератор»

При выполнении теста «Датчики давления» на экране ПК будут появляться окна аналогичные окну на рисунке 3.20. Оператору необходимо вручную устанавливать давление на калибраторе и нажимать кнопку «Продолжить».

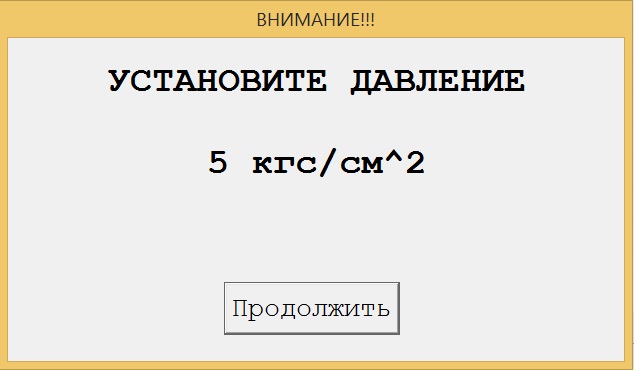


Рис. 3.20 - Окно «Установите давление»

После нажатия на кнопку продолжить появится окно, показанное на рисунке 3.21. Подобные окна будут появляться для каждого значения подаваемого давления. Закрывается окно автоматически.

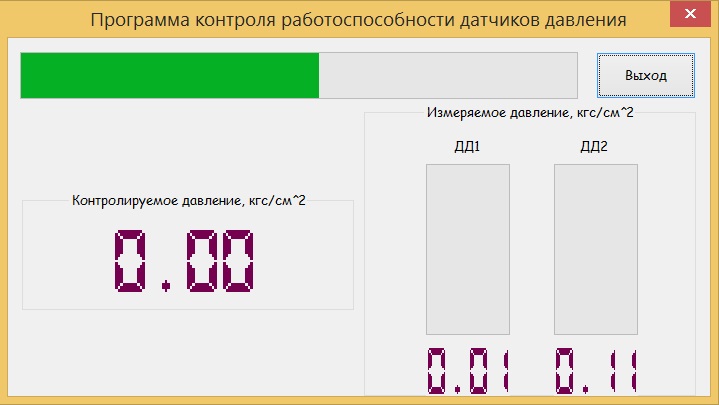


Рис. 3.21 - Окно программы контроля работоспособности датчиков давления

При проведении теста «БЧЭ-РИФ» на экране ПК появится окно, изображенное на рисунке 3.22.

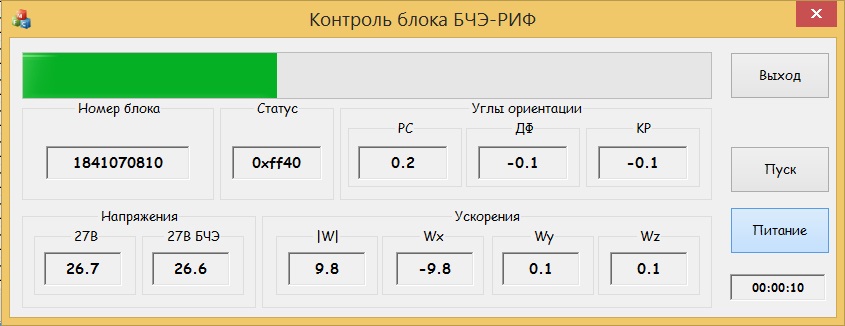


Рис. 3.22 - Окно программы контроля работоспособности блока БЧЭ-РИФ

Оператору необходимо проконтролировать на индикаторе ИП увеличение значения тока, но не более 6А.

# ЗАКЛЮЧЕНИЕ

В соответствии с поставленной задачей была разработана автоматизированная система контроля и проверки оборудования, повышающая эффективность и удобство при работе с контрольно-проверочной аппаратурой на предприятии ООО НПП «Редан». Необходимый функционал полностью реализован:

* реализован контроль аппаратуры как в ручном, так и в автоматизированном режиме за счет разделения задач между разными программами;
* организован контроль всех подсистем благодаря широкому набору приборов в стойке и разработанной библиотеке для работы с ними.
* обеспечена высокая степень модифицируемости программы за счет обеспечения проверок отдельными программами
* разработана библиотека для работы с pdf-файлами, благодаря чему имеется возможность сохранить протокол результатов контроля в защищенном виде.

Комплекс программ работает стабильно и выдает оператору понятные сообщения о результате работы, программы имеют логичную структуру, снабжены поясняющими комментариями и могут быть изменены в соответствии с требованиями заказчика.

Также в рамках работы была подготовлена необходимая сопровождающая документация, соответствующая ЕСПД.

Одним из потенциальных путей дальнейшего развития данного ПП видится добавление дополнительных проверок и аппаратуры, а также обеспечение поддержки других изделий предприятия. Благодаря удобной структуре программ их могут модифицировать другие программисты предприятия.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Программно-аппаратный комплекс для приемопередающего устройства. https://www.irz.ru/products/kpa/417.htm.
2. Кучин Н. В., Павлова М. М. Основы программирования на языке СИ: Учеб. пособие. – СПб.: СПбГУАП, 2001. – 86 с.
3. Шилдт Г. C++. Базовый курс: Пер. с англ. – М.: Вильямс, 2015. – 624 с.
4. С++. https://ru.wikipedia.org/wiki/C%2B%2B.
5. Страуструп Б. Язык программирования С++: Пер. с англ. – М.: Бином, 2015. – 1136 с.
6. Канер Сэм и др. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ. — К.: Издательство «ДиаСофт» , 2001. — 544 с.

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

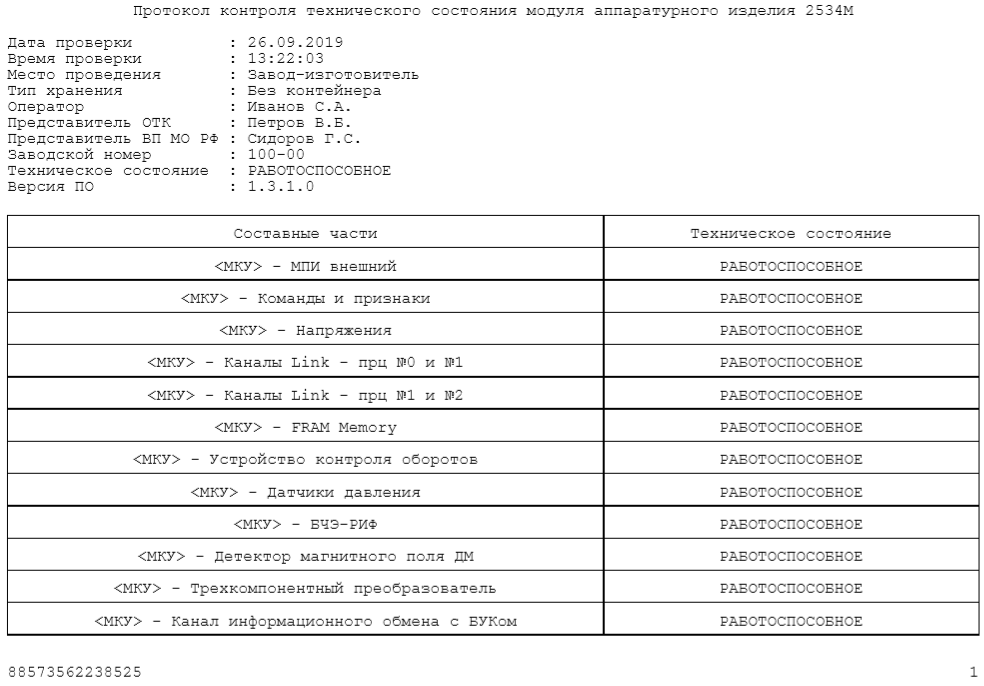
* АКПА – автоматизированная контрольно-проверочная аппаратура;
* ООО – общество с ограниченной ответственностью;
* НПП – научно-производственное предприятие;
* ПК – персональный компьютер;
* МА – модуль аппаратурный;
* ПО – программное обеспечение;
* ИРЗ – ижевский радиозавод;
* АРМ – автоматизированное рабочее место;
* NI – National Instruments;
* MFC – Microsoft Foundation Classes;
* ООП – объектно-ориентированное программирование;
* БЭ – блок электронный;
* ЭВМ – электронно-вычислительная машина;
* ОТК – отдел технического контроля;
* ВП – военное представительство;
* TCP – Transmission Control Protocol;
* МПИ – магистральный последовательный интерфейс;
* МКУ – модуль контроллеров управления;
* БЧЭ – блок чувствительных элементов;
* RS – Recommended Standard;
* ПЗУ – постоянное запоминающее устройство;
* АЦП – аналого-цифровой преобразователь
* ОУ – оконечное устройство;
* USB – Universal Serial Bus;
* МВУ – модуль внешних устройств;
* ОЗУ – оперативное запоминающее устройство;
* ППЗУ – программируемое постоянное запоминающее устройство;
* БПО – бортовое программное обеспечение;
* IP – Internet Protocol;
* MAC – Media Access Control;
* ОС – операционная система;
* MS – Microsoft;
* ГГц – гигагерц;
* Мб – мегабайт;
* Гб – гигабайт;
* МО – Министерство обороны;
* РФ – Российская Федерация;
* ИП – источник питания;
* ПП – программный продукт;
* ЕСПД – единая система программной документации.

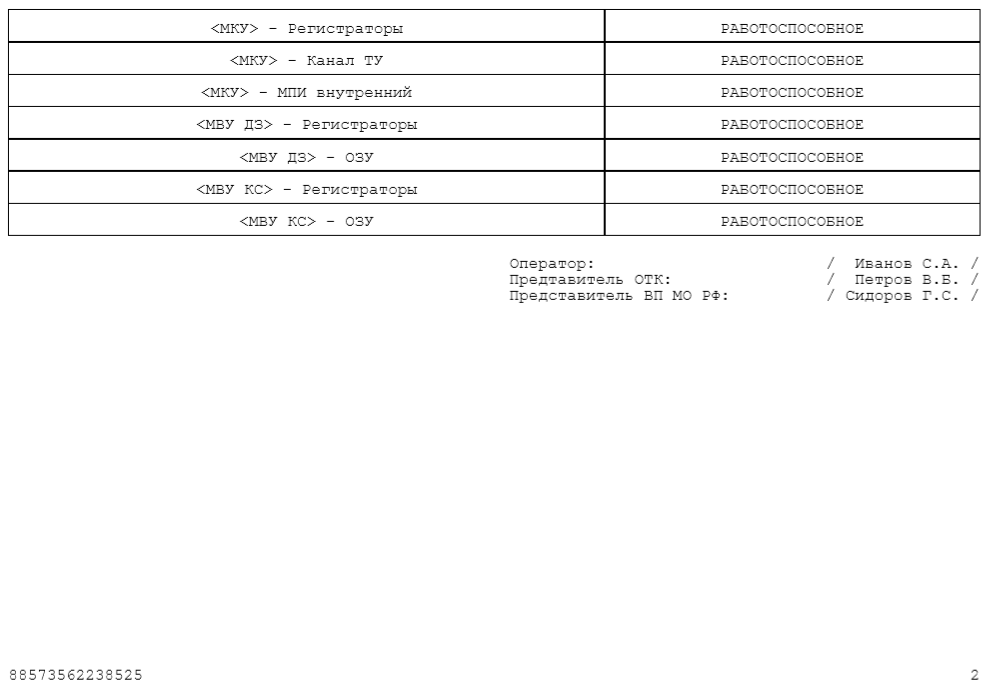
# ПРИЛОЖЕНИЯ

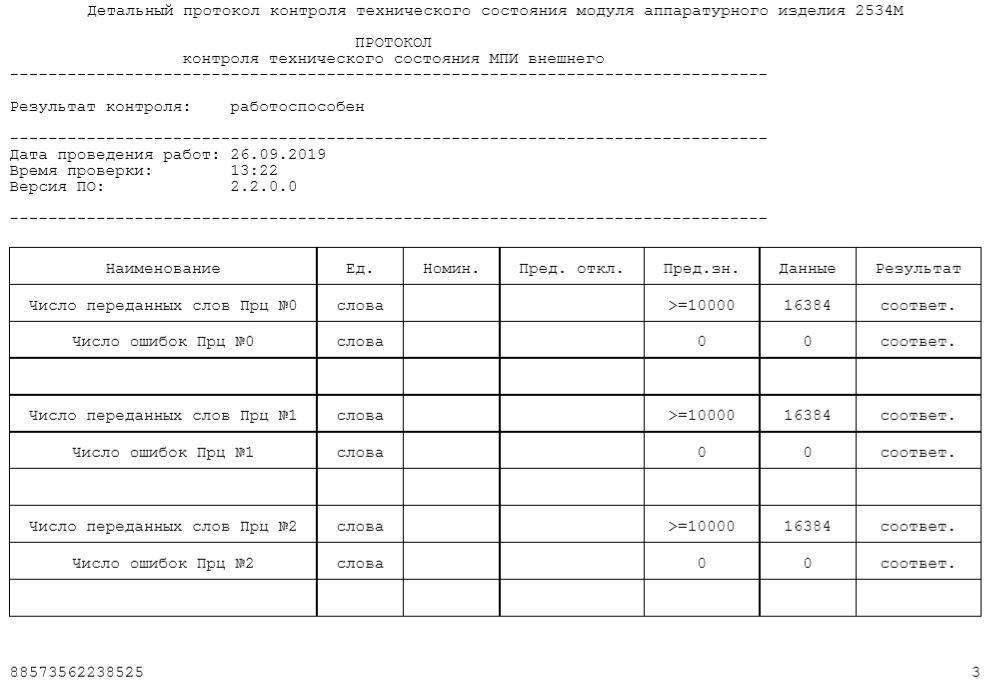
# Приложение А. Текст программы

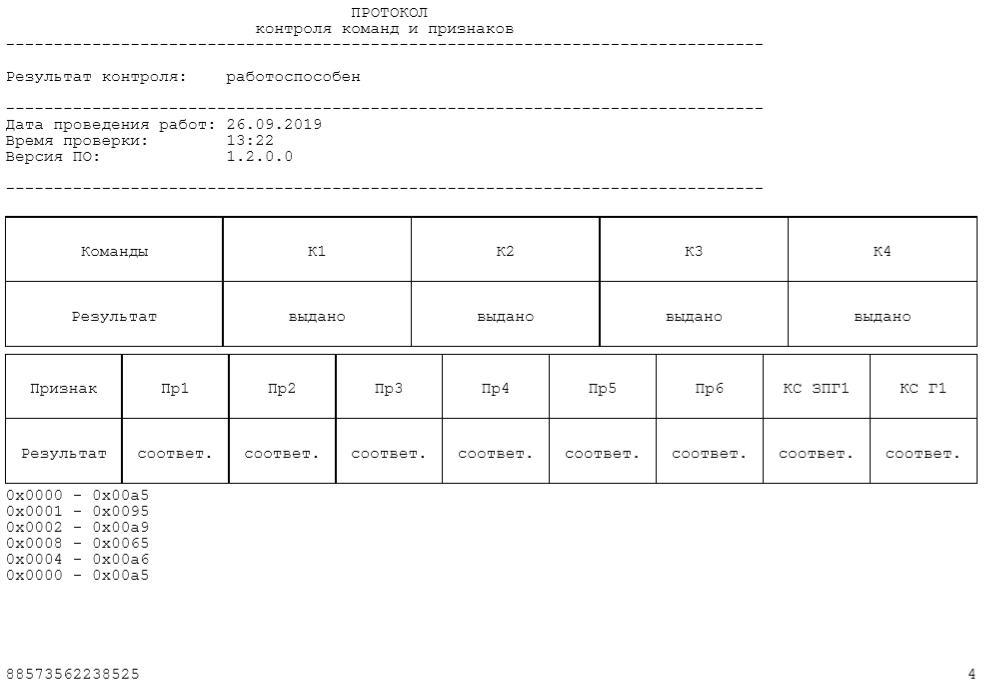
Текст программы представлен на электронном носителе.

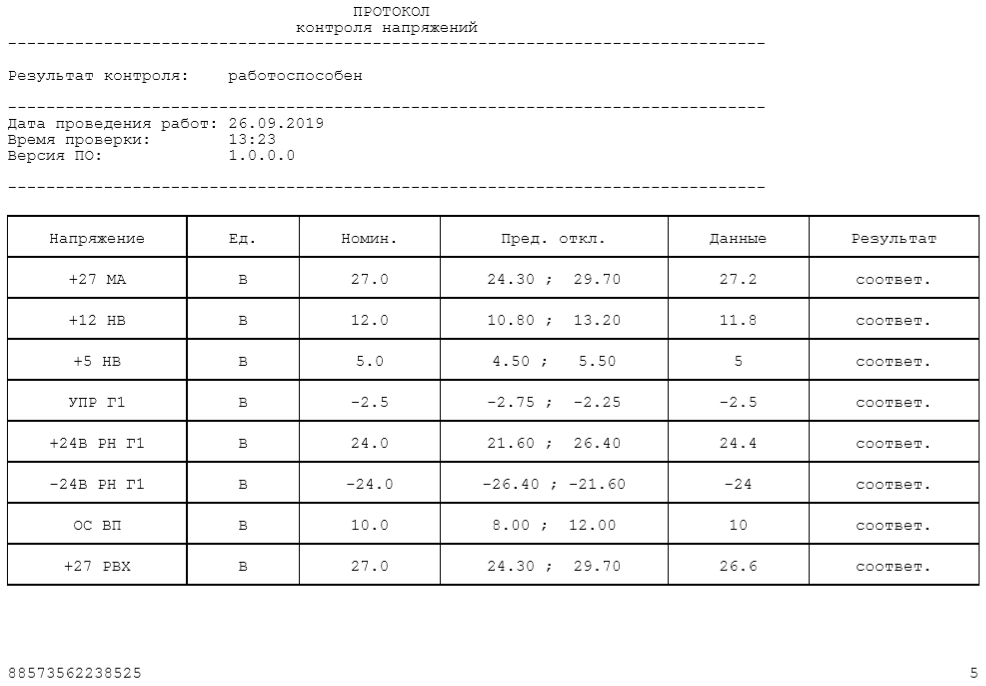
# Приложение Б. Формат протокола

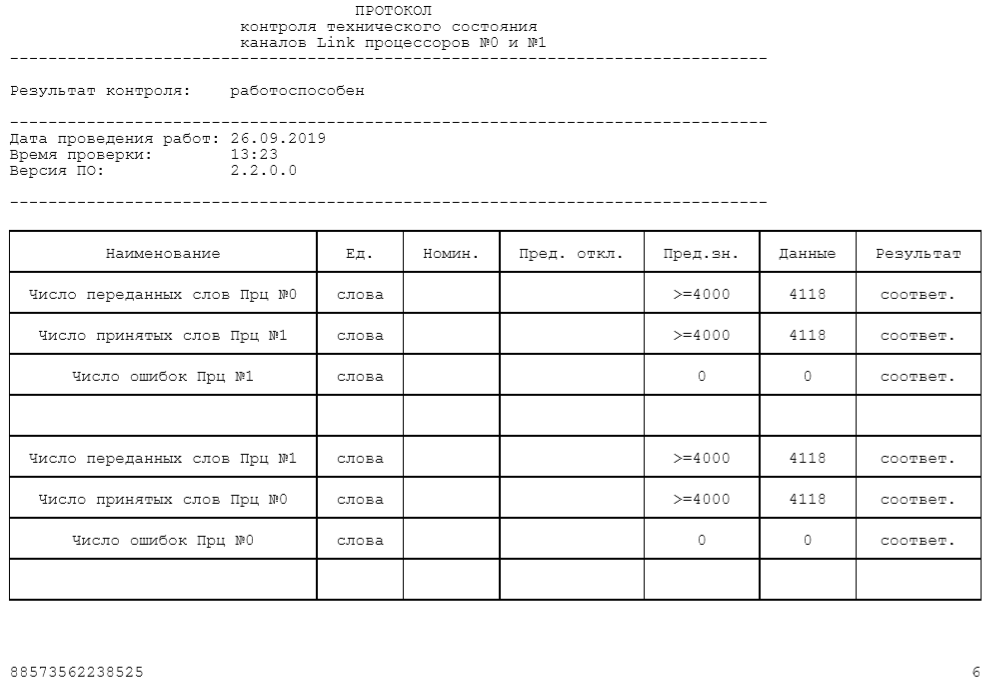
****

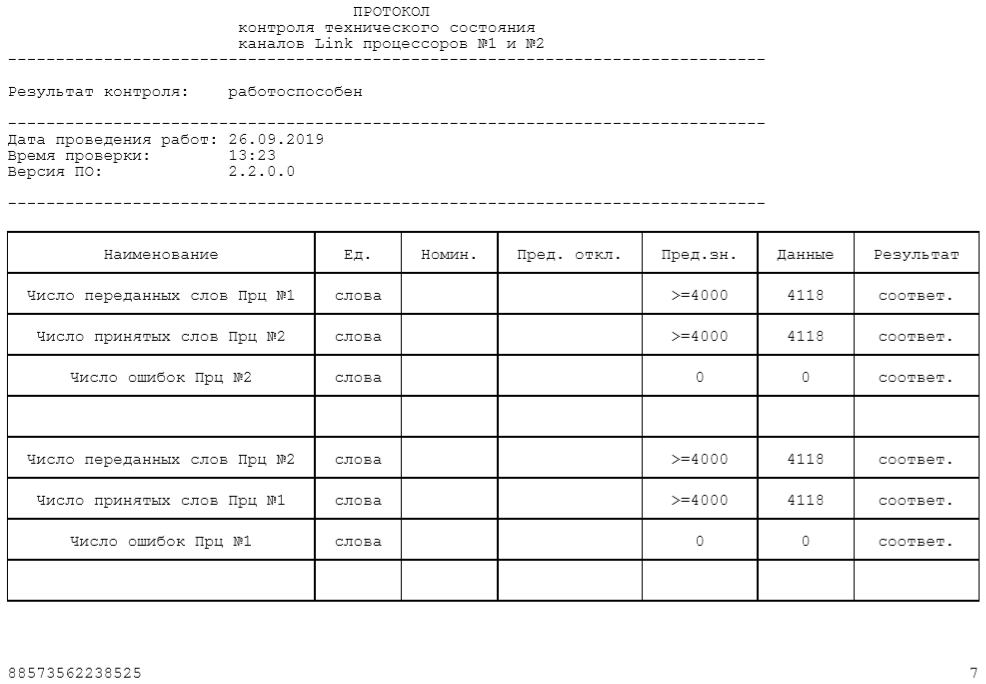
****

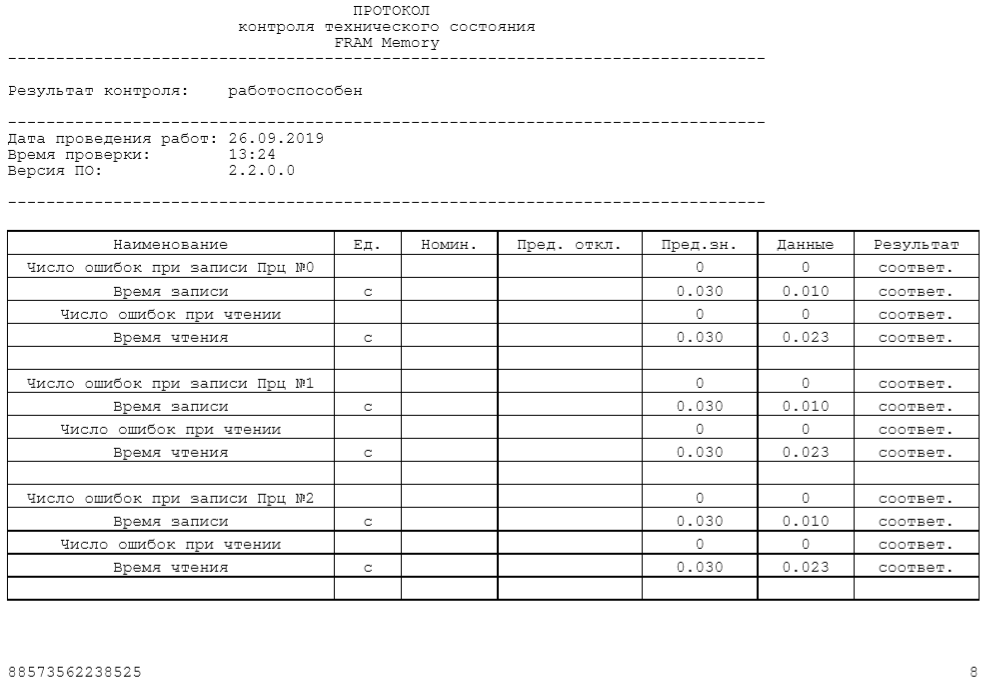
****

****

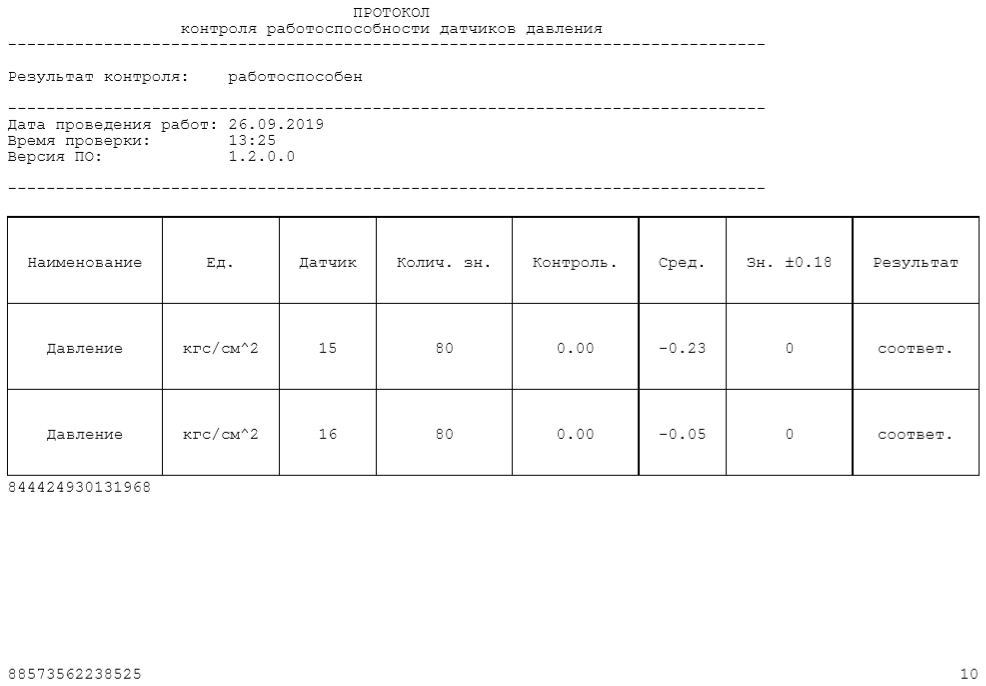
****

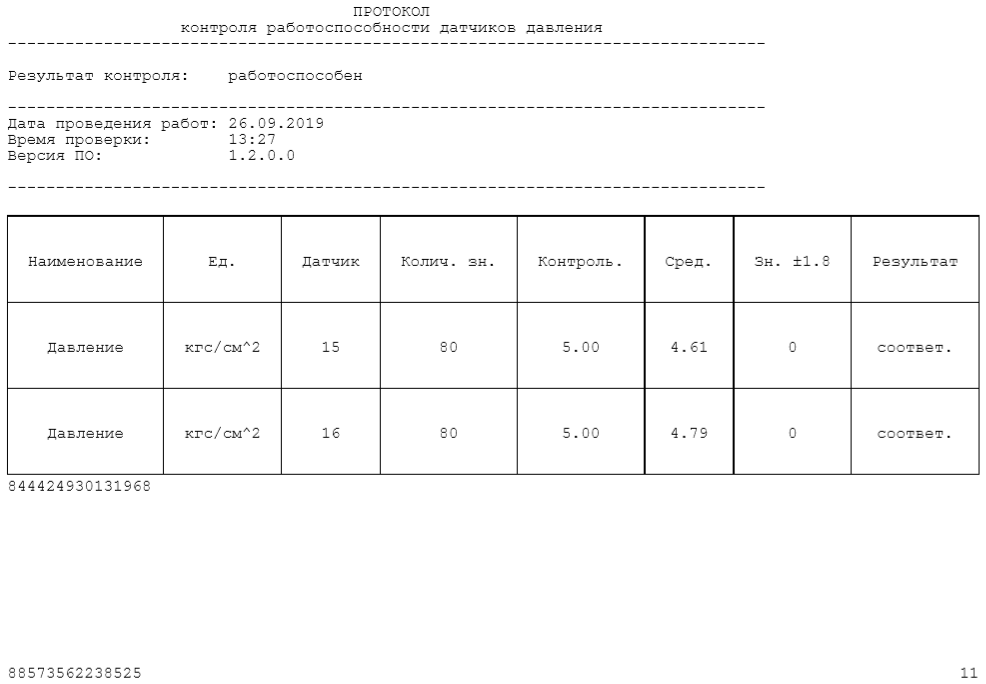
****

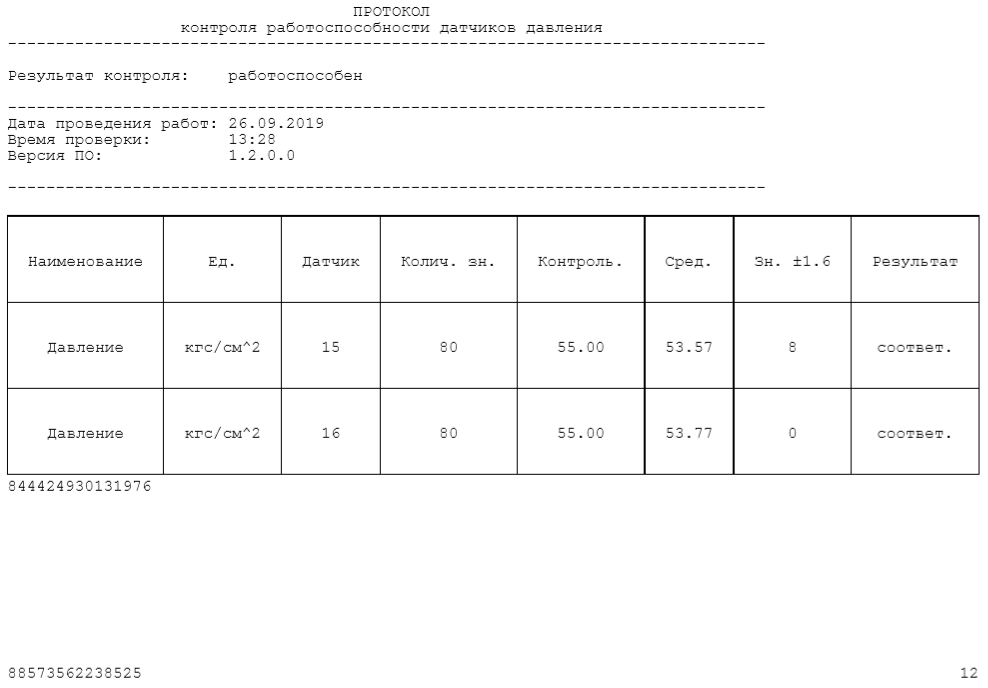
****

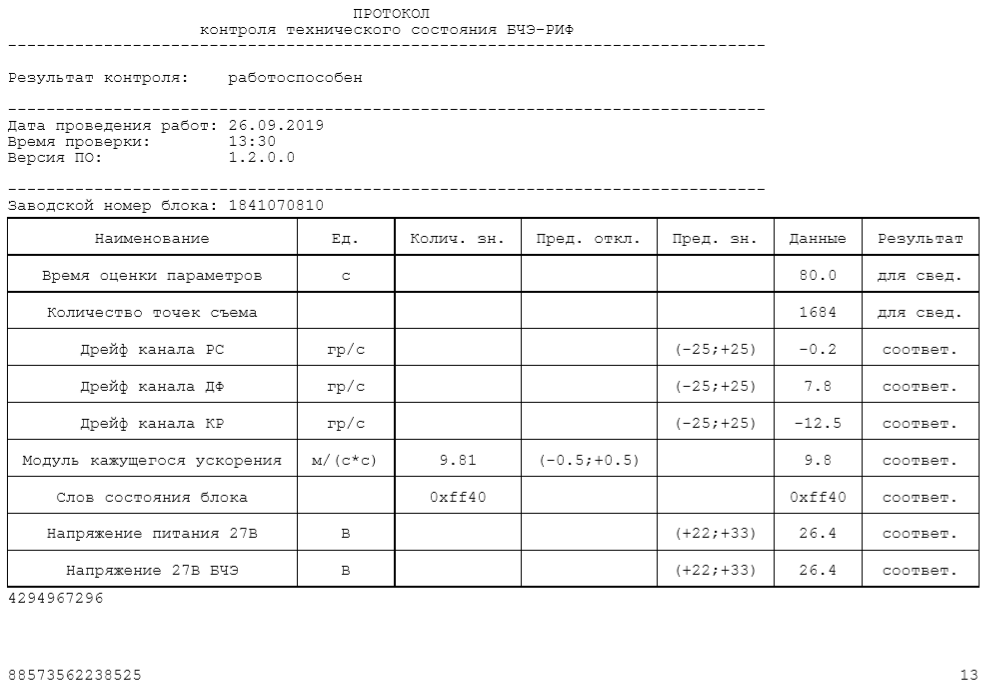
****

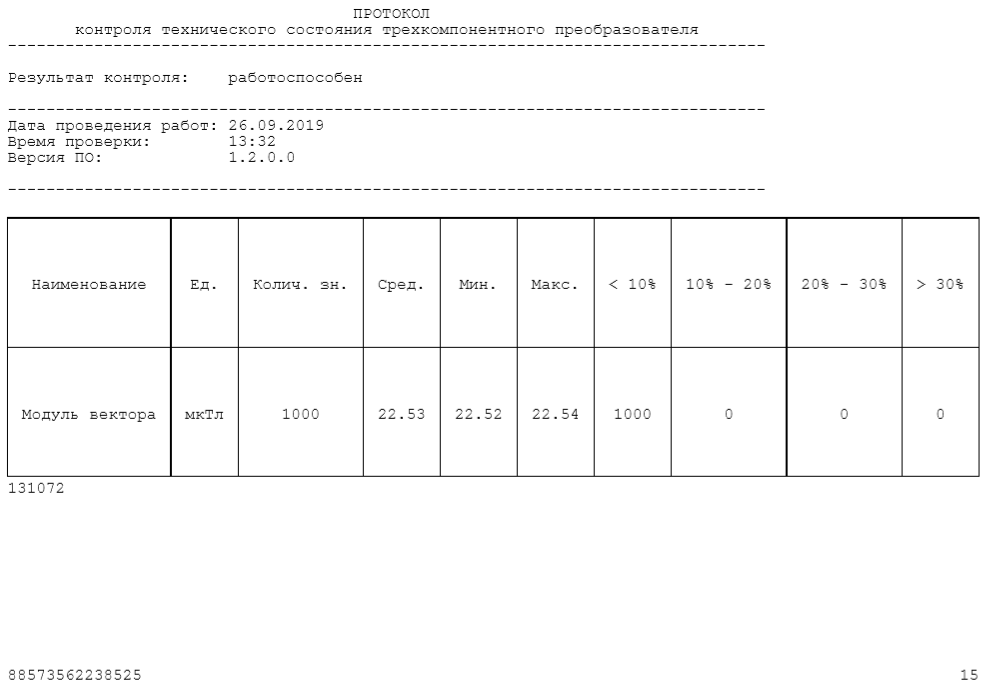
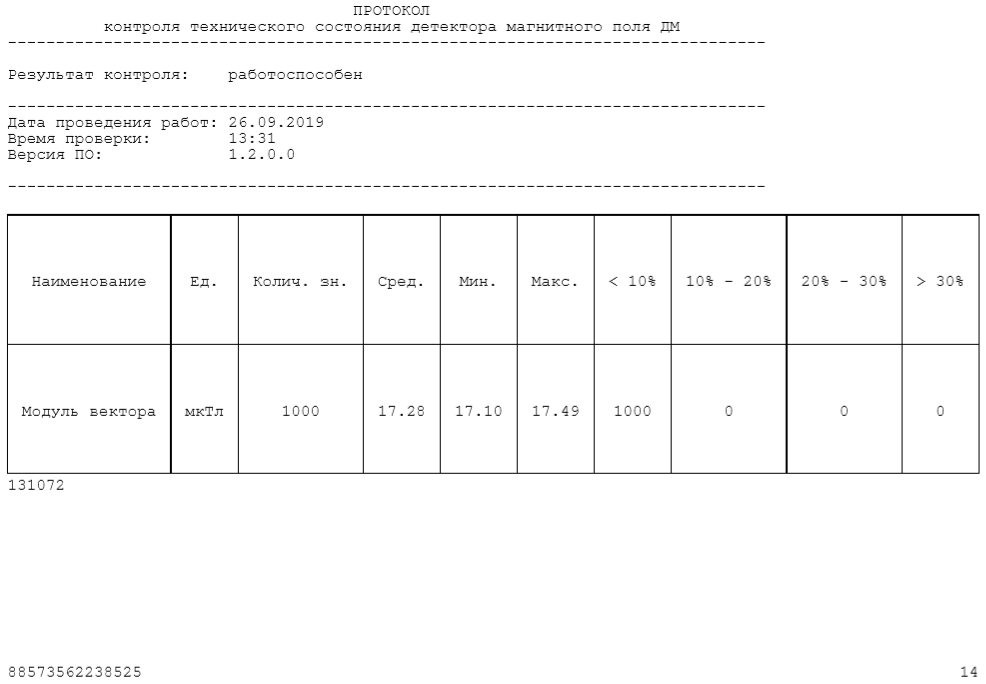
****

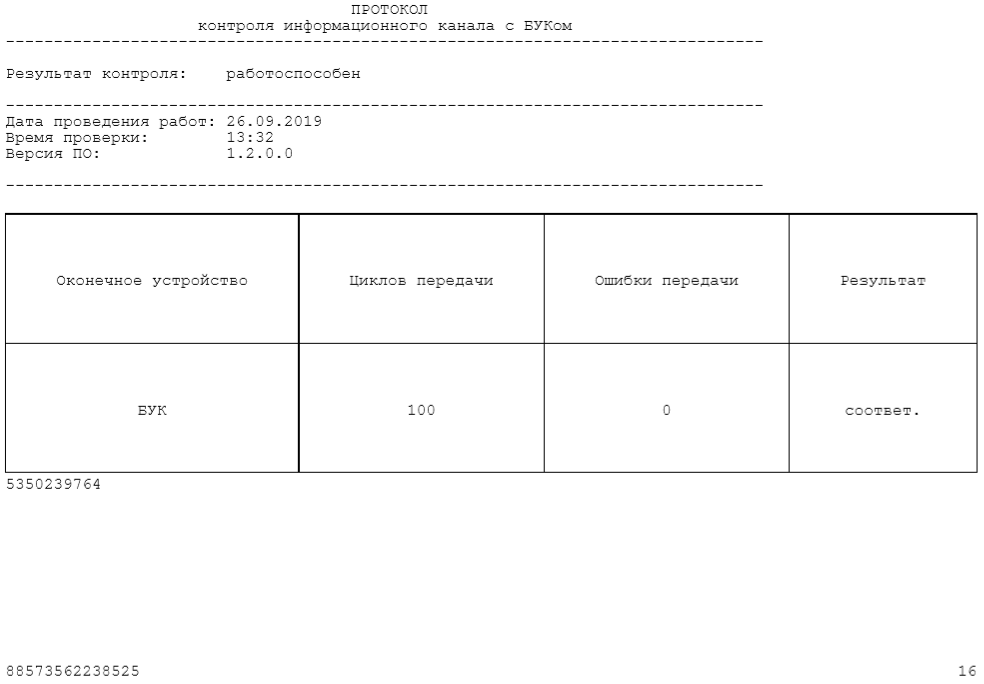
****

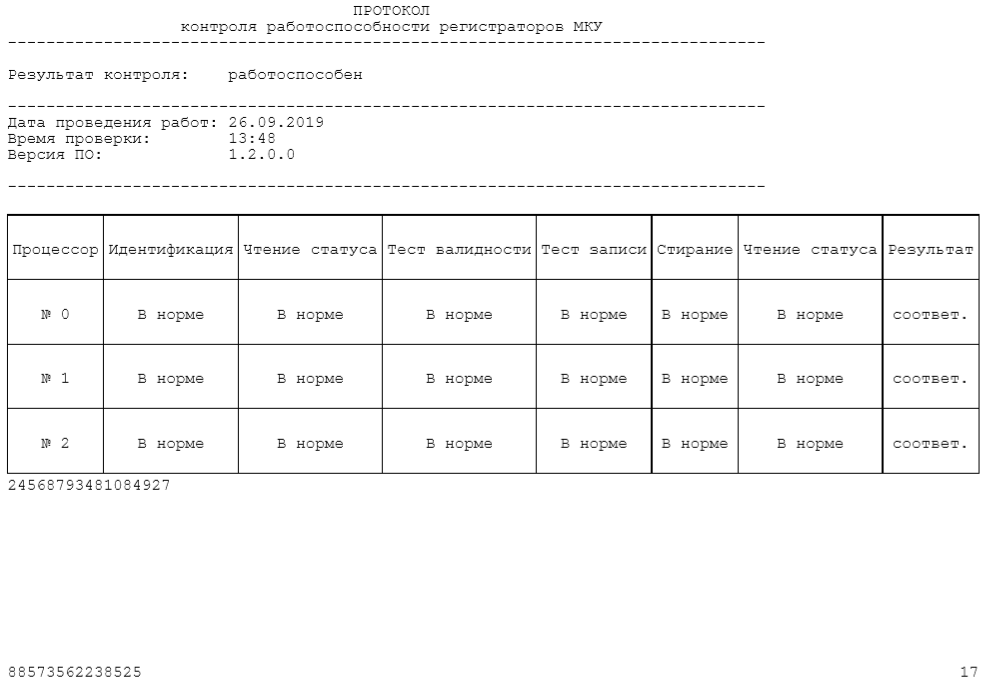
****

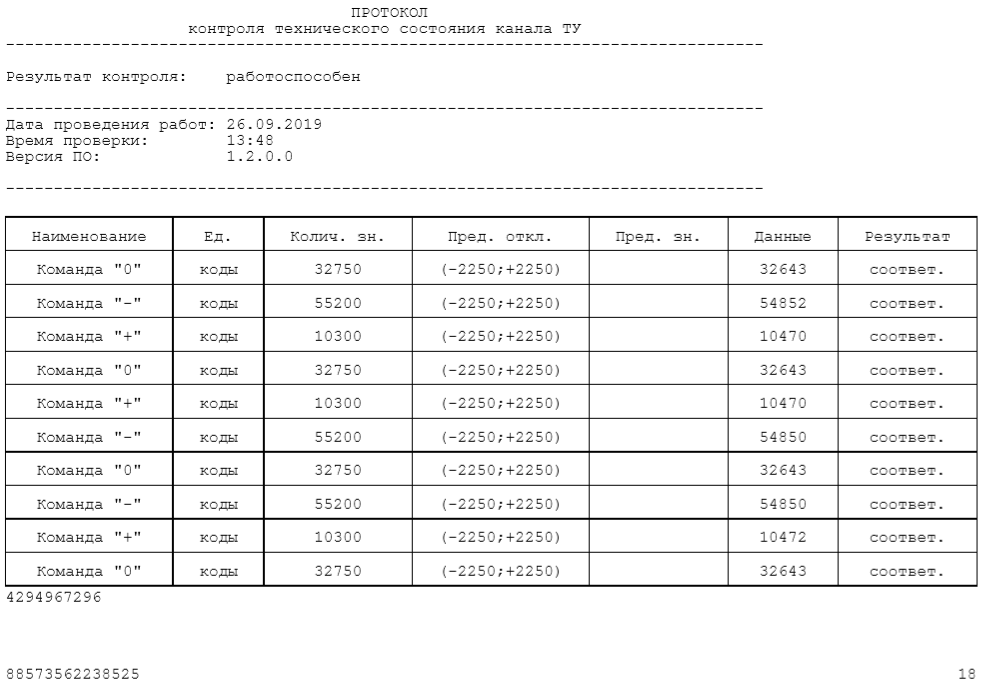
****

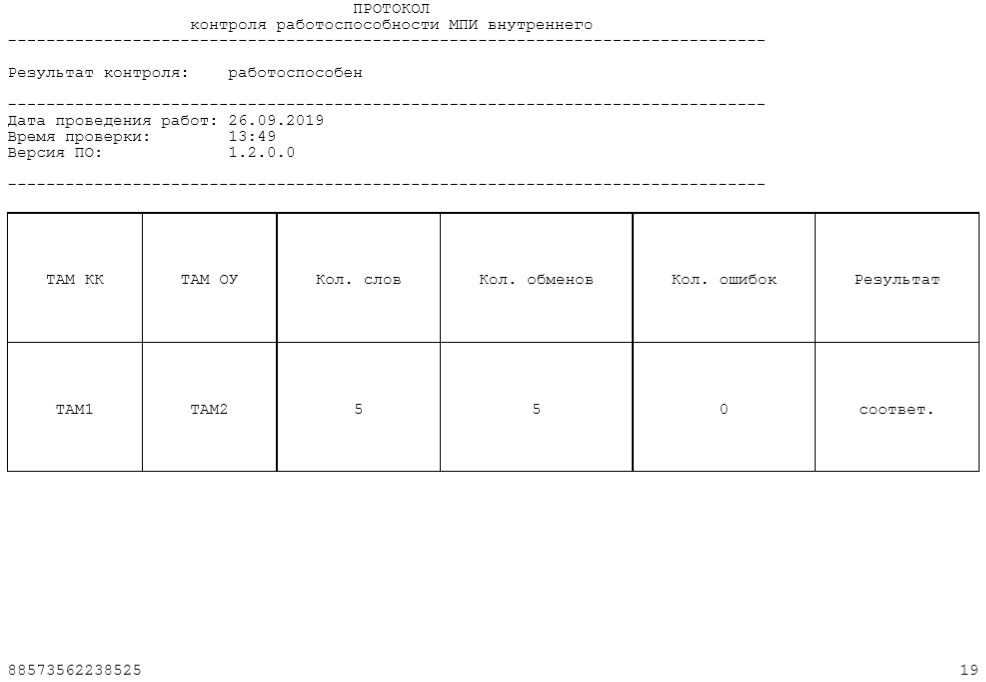
****

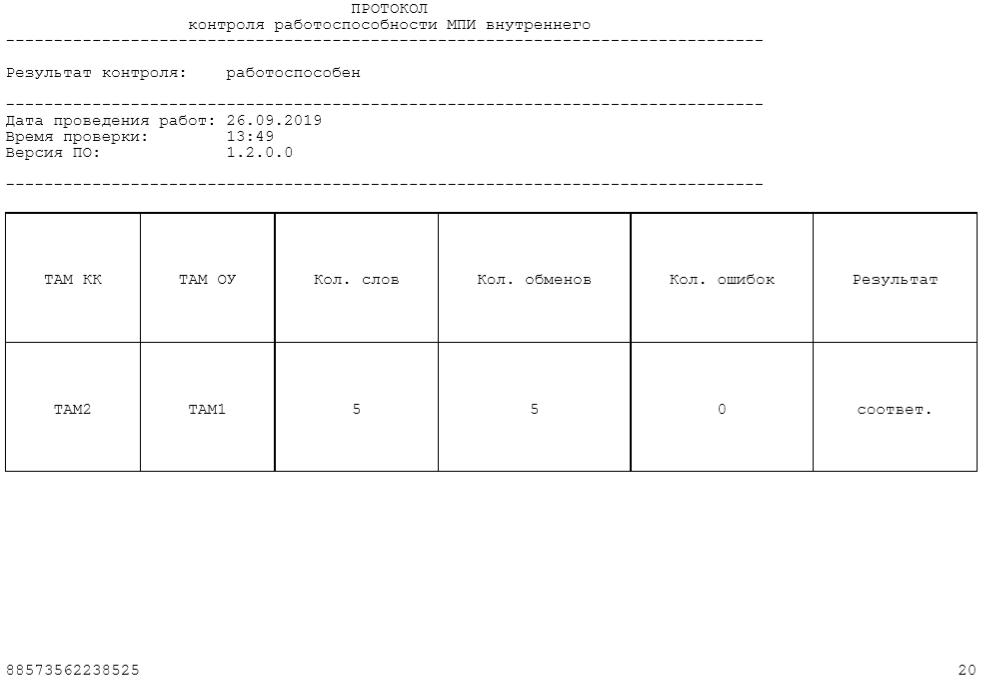
****

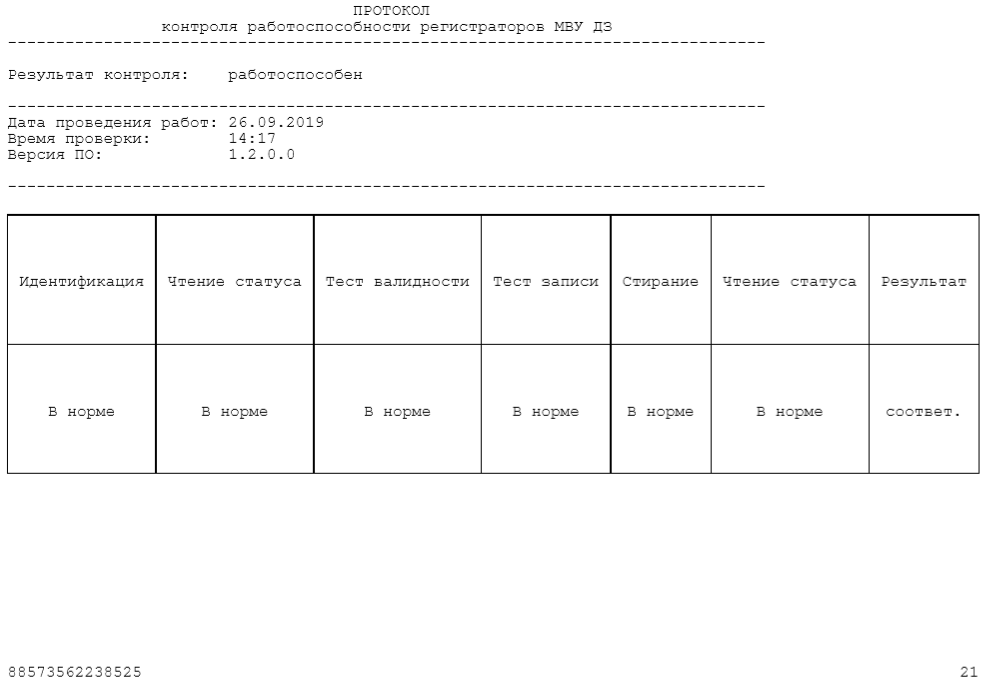
****

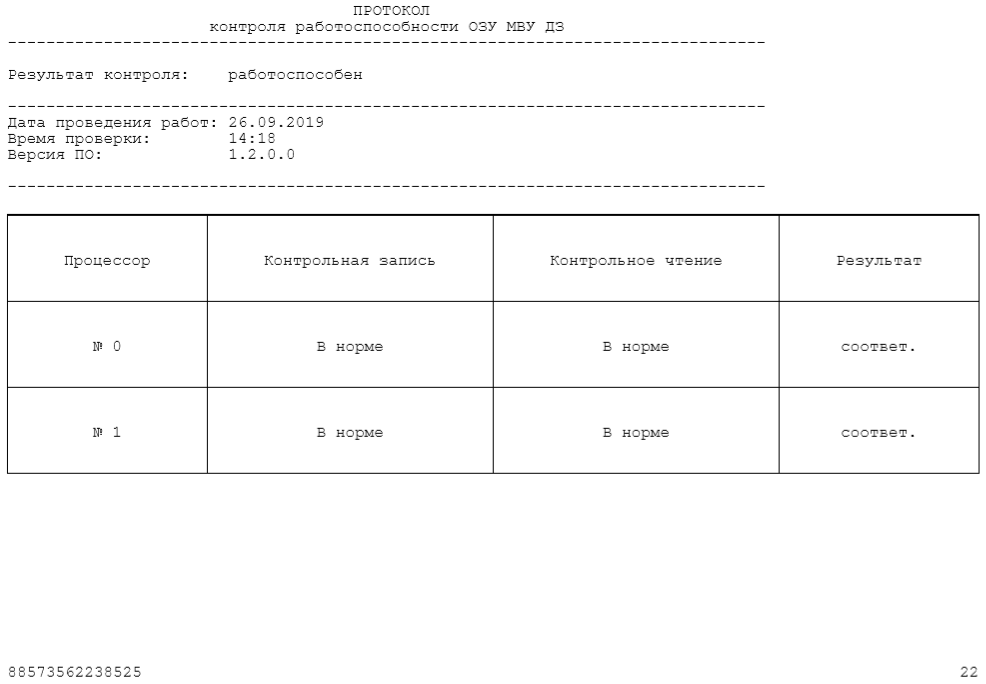
****

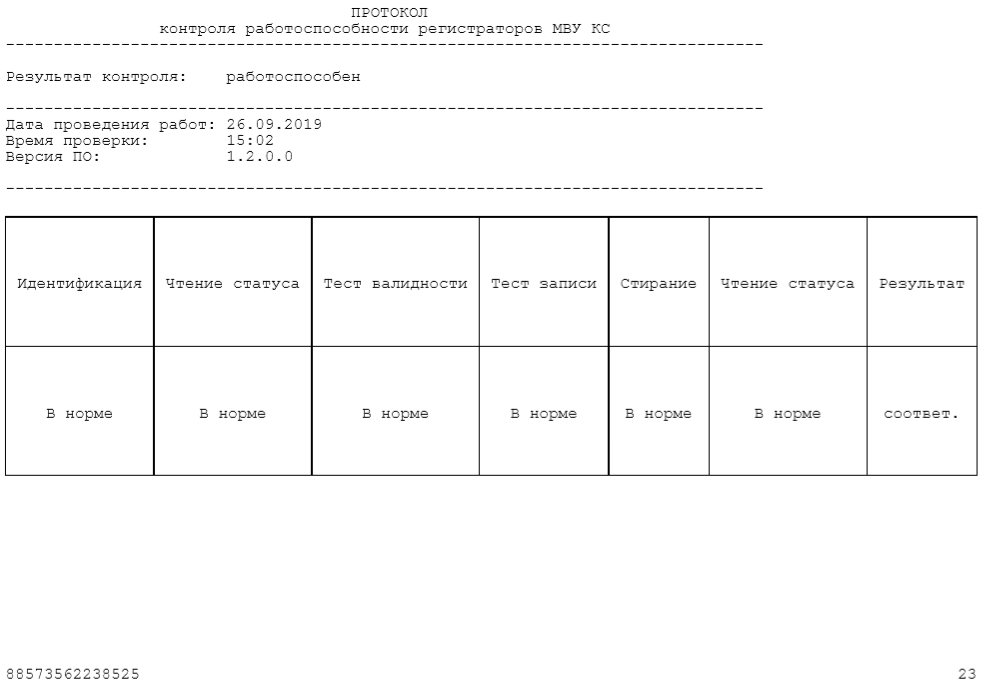
****

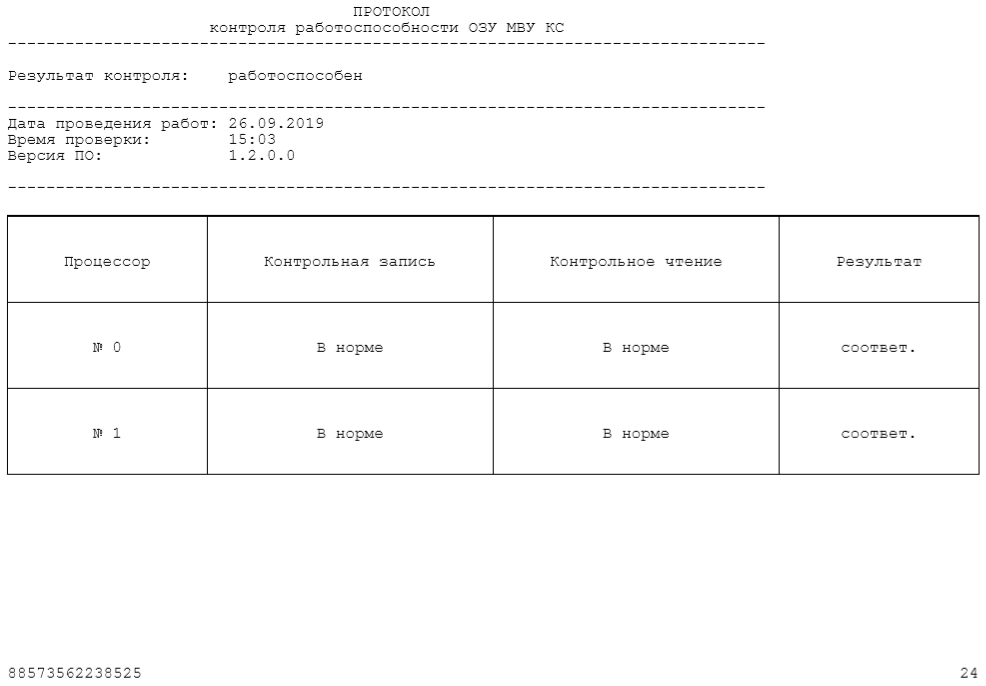
****

****

****

****

****

****

# C:\Users\1\Desktop\Диплом_маг\Сопроводительные документы\Scan — копия.jpgПриложение В. Акт о внедрении