**Описание функций и алгоритмов работы ПО «Конфигуратор»**

ПО «Конфигуратор» предназначен для выполнения следующих функций при работе с пультами ПДО:

- настройка связи с устройством;

- установка серийного номера;

- чтение ключа шифрования;

- установка ключа шифрования;

- запись конфигурации;

- чтение конфигурации;

- чтение версий выбранных устройств;

- передача прошивки в выбранные устройства.

1. **Алгоритм конфигурирования пульта ПДО**

Процесс конфигурирования пульта ПДО заключается в установке серийного номера, ключа шифрования и передачи конфигурации. Перед осуществлением процесса конфигурирования пульта ПДО необходимо настроить параметры связи ПО «Конфигуратор» согласно способу подключения устройства (RS232 или Сеть) и сформировать конфигурационный план для пульта ПДО. Далее в интерфейсе окна конфигурирования необходимо заполнить поля ввода серийного номера и ключа шифрования. Непосредственно серийный номер и ключ шифрования не входит в конфигурацию пульта, но необходимы для функционирования самого пульта ПДО так и связи с ним по каналу Ethernet. Алгоритм конфигурирования пульта ПДО заключается в последовательном выполнении следующих действий:

* установки серийного номера пульта ПДО, алгоритм которого представлен на рисунке 1.3;
* установки ключа шифрования, алгоритм которого представлен на рисунке 1.4;
* передачи конфигурации пульту ПДО, алгоритм которого представлен на рисунке 1.1.

Кроме осуществления процесса конфигурирования имеется функциональная возможность считывания ключа ПДО, алгоритм которого представлен на рисунке 1.5, и чтения конфигурации из пульта ПДО, алгоритм которого представлен на рисунке 1.2.

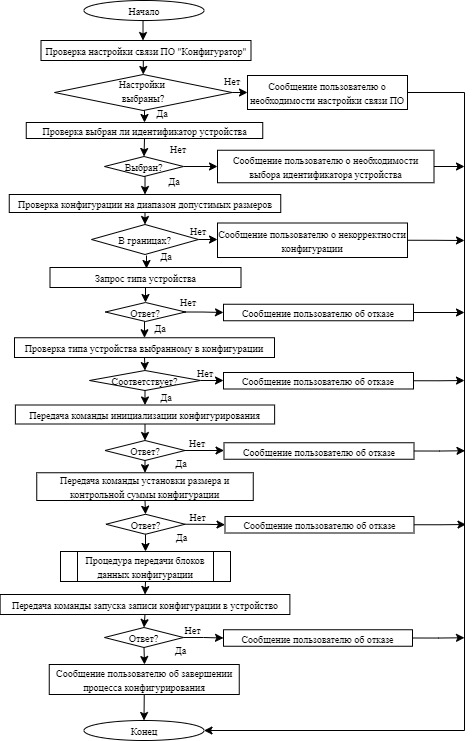


Рисунок 1.1 Процесс передачи конфигурации пульту ПДО

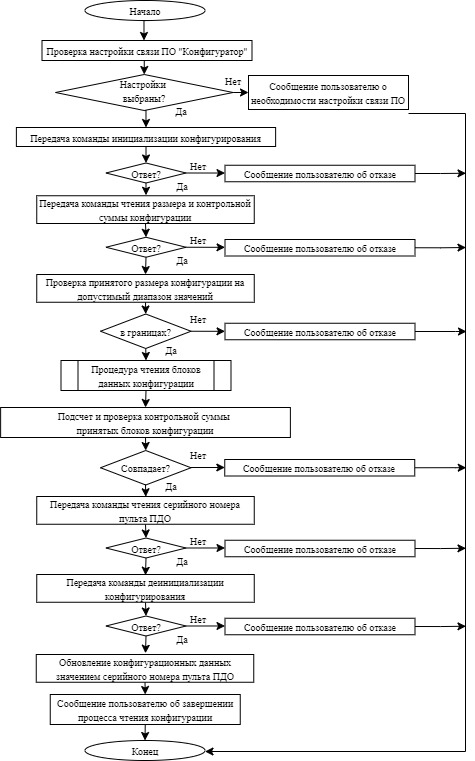


Рисунок 1.2 Процесс чтения конфигурации из пульта ПДО

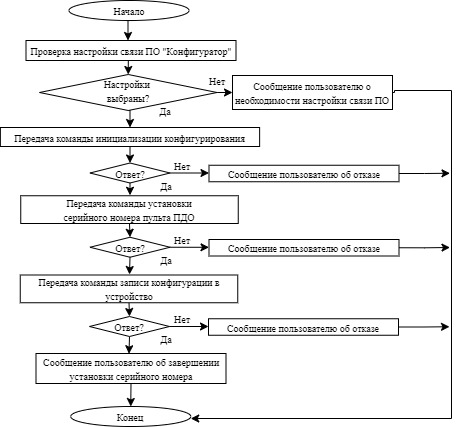


Рисунок 1.3 Процесс установки серийного номера пульта ПДО

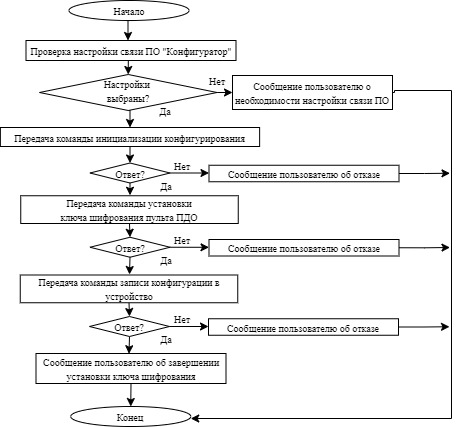


Рисунок 1.4 Процесс установки ключа шифрования пульта ПДО

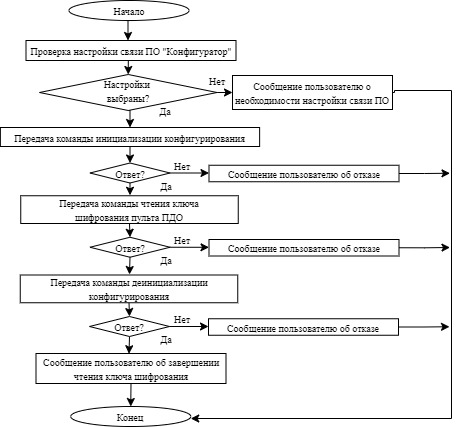


Рисунок 1.5 Процесс чтения ключа шифрования пульта ПДО

1. **Алгоритм обновления ПО пульта ПДО**

Процесс обновления ПО пульта ПДО заключается в передаче файла образа программы и установки версии ПО в пульт ПДО. Перед осуществлением процесса обновления ПО пульта ПДО необходимо настроить параметры связи ПО «Конфигуратор» согласно способу подключения устройства (RS232 или Сеть) и сформировать конфигурационный план для пульта ПДО. Далее в интерфейсе окна обновления ПО необходимо заполнить поля ввода версии ПО и ключа шифрования.

Алгоритм обновления ПО пульта ПДО представлен на рисунке 2.1.

Кроме осуществления процесса обновления ПО пульта ПДО имеется функциональная возможность считывания записанной версии ПО, алгоритм которого представлен на рисунке 2.2.

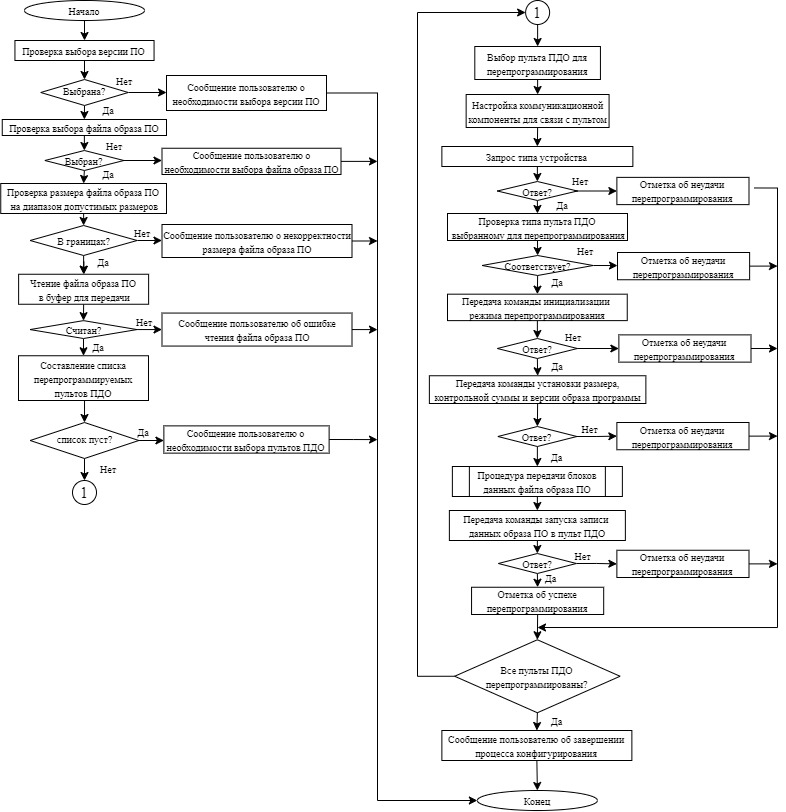
****

Рисунок 2.1 Процесс обновления ПО пульта ПДО

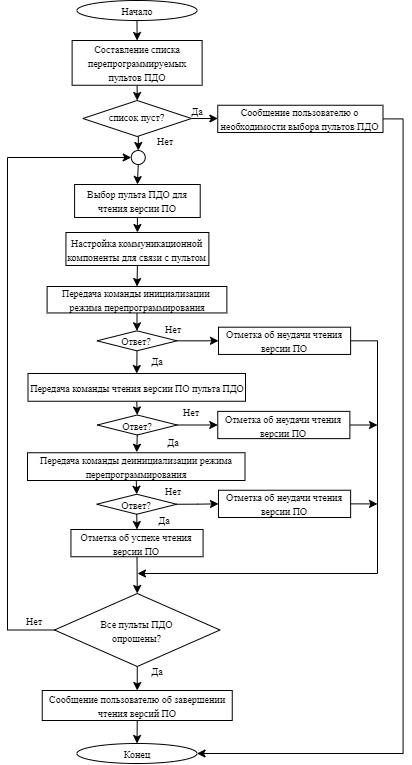
****

Рисунок 2.2 Процесс чтения записанной версии ПО пульта ПДО

1. **Список модулей ПО «Конфигуратор»**

ПО «Конфигуратор» содержит ниже перечисленные модули:

* Модуль настройки соединения. Содержит в себе класс (SetupConnectionsDialog), осуществляющий функции отображения, выбора параметров связи с подключенным пультом ПДО;
* Модуль конфигурирования. Содержит класс (ConfigureDialog), осуществляющий функции приема/передачи сформированной конфигурации пульту ПДО, установку серийного номера, установку/чтение ключа шифрования;
* Модуль обновления ПО. Содержит класс (FmWareDialog), осуществляющий функции обновления ПО пульта ПДО, опроса версий ПО, подключенных пультов ПДО;
* Модуль взаимодействия с RS232 и Ethernet. Содержит класс (CommunicationDialog) для приема/передачи данных прошивки и конфигурации изделия по установленному каналу связи;
* Модуль шифрования/дешифрования данных передаваемых по каналу Ethernet (CryptoDialog);
* Модуль отладки. Содержит класс (DebugViewDialog), осуществляющий включение/выключение режима просмотра отладочных сообщений;
* Модуль сервиса. Содержит класс (ServiceViewDialog), осуществляющий включение/выключение режима просмотра сервисных сообщений.

1. **Описание классов ПО «Конфигуратор»**

4.1 CommunicationDialog

Класс предназначен для взаимодействия интерфейса пользователя с подключенным устройством для осуществления процесса конфигурирования и обновления ПО. Класс содержит следующие функции:

* setMode – устанавливает режим соединения с устройством (RS232 или Ethernet);
* setAesKey – устанавливает значение ключа шифрования для передачи/приема данных устройству по каналу Ethernet;
* setSelfIP – устанавливает значение IP адреса ПЭВМ для передачи/приема данных по каналу Ethernet (вызывается в случае установки режима связи - Ethernet);
* setTargetIP – устанавливает значение IP адреса подключенного устройства (вызывается в случае установки режима связи - Ethernet);
* Open – попытка открытия и настройки канала связи с устройством;
* Close – закрытие канала связи с устройством;
* Add2Log – добавление сообщения при приеме/передачи в специальный файл в целях отладки связи с устройством (технологическая функция);
* getLogTxStr – чтение последней переданной команды устройству (технологическая функция);
* getLogRxStr - чтение последней принятого ответа от устройства (технологическая функция);
* Data2Hex – функция преобразования массива данных в их Hex-представление. Используется для формирования текстовой строки-команды устройству;
* setRS232PortName – устанавливает имя последовательного порта для связи с устройством (вызывается в случае установки режима связи – RS232);
* getLastErrorString - чтение последнего ошибочного ответа/отсутствие ответа от устройства;
* crc32 – инициализация массива данных для подсчета контрольной суммы блока данных;
* ProcessCRC – формирование значения контрольной суммы блока данных;
* fnGetCFGUI – функция запроса идентификатора типа подключенного устройства;
* fnConfigOn – функция инициализации режима конфигурирования устройства;
* fnConfigOff – функция деинициализации режима конфигурирования устройства;
* fnConfigSetSysInfo – функция установки системных данных конфигурации (размер, контрольная сумма, версия);
* fnConfigGetSysInfo – функция чтения системных данных конфигурации (размер, контрольная сумма, версия);
* fnConfigSetDataSize – функция установки размера конфигурации;
* fnConfigGetDataSize – функция чтения размера конфигурации;
* fnConfigSetCRC – функция установки контрольной суммы конфигурации;
* fnConfigGetCRC – функция чтения контрольной суммы конфигурации;
* fnConfigSetVersion – функция установки версии конфигурации;
* fnConfigGetVersion – функция чтения версии конфигурации;
* fnConfigSetBlock – функция передачи блока данных конфигурации;
* fnConfigGetBlock – функция чтения блока данных конфигурации;
* fnConfigStartWrite – функция запуска перезаписи конфигурации из ОЗУ устройства в EEPROM;
* fnFwwareOn – функция инициализации режима обновления ПО устройства;
* fnFwwareOff – функция деинициализации режима обновления ПО устройства;
* fnFwwareSetSysInfo – функция установки системных данных образа программы (размер, контрольная сумма, версия);
* fnFwwareGetSysInfo – функция чтения системных данных образа программы (размер, контрольная сумма, версия);
* fnFwwareSetDataSize – функция установки размера образа программы;
* fnFwwareGetDataSize – функция чтения размера образа программы;
* fnFwwareSetCRC – функция установки контрольной суммы образа программы;
* fnFwwareGetCRC – функция чтения контрольной суммы образа программы;
* fnFwwareSetVersion – функция установки версии образа программы;
* fnFwwareGetVersion – функция чтения версии образа программы;
* fnFwwareSetBlock – функция передачи блока данных образа программы;
* fnFwwareGetBlock – функция чтения блока данных образа программы;
* fnFwwareStartWrite – функция запуска перезаписи образа программы из ОЗУ устройства во Flash память;
* fnSetKeyAes – функция установки/смены ключа шифрования в устройстве;
* fnGetKeyAes – функция чтения ключа шифрования записанного в устройстве;
* fnSetSN – функция установки/смены серийного номера в устройстве;
* fnGetSN – функция чтения серийного номера записанного в устройстве;
* serialPort\_dataRead – функция-обработчик данных принятых из последовательного порта;
* pTimeTimeouts\_timeout – функция обработки таймера ожидания ответа от устройства;
* readPendingDatagrams – функция чтения данных принятых по каналу Ethernet;
* parse\_msg – функция декодирования принятого сообщения от устройства;
* setRqSetBlock – функция передачи устройству блока данных;
* setRqGetBlock – функция чтения блока данных из устройства;
* processTheDatagram – функция декодирования данных принятых по каналу Ethernet;
* cancel\_transactions – функция обработки сигнала отказа в выполнении текущей транзакции.

4.2 CryptoDialog

Класс предназначен для шифрования/дешифрования блока данных. Класс содержит следующие функции:

* setKey – функция установки значения ключа шифрования;
* EncodeBuffer – функция шифрования буфера данных;
* DecodeBuffer – функция дешифрования буфера данных;
* KeyInit – функция инициализации ключа шифрования;
* Code\_Block – функция шифрования блока из 16-ти байт;
* Decode\_Block – функция дешифрования блока из 16-ти байт.

4.3 DebugViewDialog

Класс предназначен включения/выключения и просмотра принятых сообщений в режиме отладки. Класс содержит следующие функции:

* setTitle – установка текста заголовка формы диалога;
* setMode - устанавливает режим соединения с устройством (RS232 или Ethernet);
* setRS232Name - устанавливает имя последовательного порта для связи с устройством (вызывается в случае установки режима связи – RS232);
* setSelfIP – устанавливает значение IP адреса ПЭВМ для передачи/приема данных по каналу Ethernet (вызывается в случае установки режима связи - Ethernet);
* setAesKey – устанавливает значение ключа шифрования для передачи/приема данных устройству по каналу Ethernet;
* setConfigureStr – функция установки конфигурационного плана списка устройств;
* appendMsg – функция добавления принятого сообщения в интерфейс для отображения;
* Data2Hex – функция преобразования массива данных в их Hex-представление. Используется для формирования текстовой строки-команды устройству;
* fnDebugOn – функция включения режима приема отладочных сообщений из устройства;
* fnDebugOff – функция выключения режима приема отладочных сообщений из устройства;
* b\_start\_clicked – функция обработчик нажатия кнопки включения/выключения режима Debug устройства;
* b\_savelog\_clicked – функция обработчик нажатия кнопки сохранения, принятых ранее, отладочных сообщений;
* b\_loadlog\_clicked – функция обработчик нажатия кнопки чтения, сохраненных ранее, отладочных сообщений;
* cb\_liststations\_currentIndexChanged – функция обработчик выбора устройства из списка, установленных конфигурационным планом;
* serialPort\_dataRead – функция-обработчик данных принятых из последовательного порта;
* pTimeTimeouts\_timeout – функция обработки таймера ожидания ответа от устройства;
* readPendingDatagrams – функция данных, принятых по каналу Ethernet;
* processTheDatagram – функция декодирования данных, принятых по каналу Ethernet;
* getIPFromCid – функция вычисления IP адреса устройства по выбранному идентификатору.

4.4 ServiceViewDialog

Класс предназначен включения/выключения и просмотра принятых сообщений в режиме сервиса. Класс содержит следующие функции:

* setTitle – установка текста заголовка формы диалога;
* setMode - устанавливает режим соединения с устройством (RS232 или Ethernet);
* setRS232Name - устанавливает имя последовательного порта для связи с устройством (вызывается в случае установки режима связи – RS232);
* setSelfIP – устанавливает значение IP адреса ПЭВМ для передачи/приема данных по каналу Ethernet (вызывается в случае установки режима связи - Ethernet);
* setAesKey – устанавливает значение ключа шифрования для передачи/приема данных устройству по каналу Ethernet;
* setConfigureStr – функция установки конфигурационного плана списка устройств;
* appendMsg – функция добавления принятого сообщения в интерфейс для отображения;
* Data2Hex – функция преобразования массива данных в их Hex-представление. Используется для формирования текстовой строки-команды устройству;
* fnServiceOn – функция включения режима приема сервисных сообщений из устройства;
* fnServiceOff – функция выключения режима приема сервисных сообщений из устройства;
* b\_start\_clicked – функция обработчик нажатия кнопки включения/выключения режима Debug устройства;
* b\_savelog\_clicked – функция обработчик нажатия кнопки сохранения, принятых ранее, отладочных сообщений;
* b\_loadlog\_clicked – функция обработчик нажатия кнопки чтения, сохраненных ранее, отладочных сообщений;
* cb\_liststations\_currentIndexChanged – функция обработчик выбора устройства из списка, установленных конфигурационным планом;
* serialPort\_dataRead – функция-обработчик данных принятых из последовательного порта;
* pTimeTimeouts\_timeout – функция обработки таймера ожидания ответа от устройства;
* readPendingDatagrams – функция чтения данных, принятых по каналу Ethernet;
* processTheDatagram – функция декодирования данных, принятых по каналу Ethernet;
* getIPFromCid – функция вычисления IP адреса устройства по выбранному идентификатору.

4.5 SetupConnectionsDialog

Класс предназначен настройки параметров связи с устройством. Класс содержит следующие функции:

* instanceNetworkList – функция составления списка доступных последовательных портов или сетевых интерфейсов в зависимости от типа канала связи(RS232 или Сеть);
* insatnceEnable – настройка интерфейса программы в случае использования канала связи с устройством - RS232;
* setConnectionsParamStr – функция установки в интерфейсе параметра связи с устройством;
* getConnectionsParamStr – функция чтения из интерфейса выбранного параметра связи с устройством;
* IP2Str\_Dec – функция преобразования IP адреса в его строковое представление.

4.6 ConfigureDialog

Класс предназначен для осуществления процесса конфигурирования устройства. Класс содержит следующие функции:

* sigCancel – функция сигнал отмены процесса чтения/записи конфигурационных данных в устройство;
* setTitle – установка текста заголовка формы диалога;
* setMode - устанавливает режим соединения с устройством (RS232 или Ethernet);
* setRS232Name - устанавливает имя последовательного порта для связи с устройством (вызывается в случае установки режима связи – RS232);
* setSelfIP – устанавливает значение IP адреса ПЭВМ для передачи/приема данных по каналу Ethernet (вызывается в случае установки режима связи - Ethernet);
* setConfigureStr – функция установки конфигурационного плана списка устройств;
* getConfigureStr – функция чтения считанной конфигурации в интерфейс ПО «Конфигуратора»;
* isAcceptLoaddedConfigure – функция подтверждения применения считанной конфигурации из устройства;
* b\_open\_clicked – функция обработчик кнопки просмотра списка идентификаторов устройств;
* b\_send\_clicked – функция обработчик кнопки передачи конфигурации в устройство;
* b\_rcv\_clicked – функция обработчик кнопки чтения конфигурации из устройства;
* b\_SetSN\_clicked – функция обработчик кнопки установки серийного номера устройства;
* b\_GetAES\_clicked – функция обработчик кнопки чтения ключа шифрования, записанного в устройство;
* b\_SetAES\_clicked – функция обработчик кнопки установки ключа шифрования в устройство;
* pdSendConfig\_cancel – функция обработчик кнопки отмены процесса чтения/записи конфигурации в устройство;
* closeEvent – функция обработчик сигнала закрытия окна конфигурирования;
* ConvertToConfigureStr – функция формирования конфигурации устройства из конфигурационного плана;
* getIPFromCid – функция вычисления IP адреса из идентификатора устройства;
* addGroupCallForGid – функция формирования секции групповых вызовов устройства;
* getStationInfoFromHash – функция вычисления конфигурации из конфигурационного плана согласно hash-суммы конфигурации устройства;
* checkGidInGroupCallList – проверка нахождения выбранного устройства в списке групповых вызовов конфигурационного плана;
* updateSNInRcvConfig – функция обновления серийного номера в считанной конфигурации из устройства;
* getLN\_SN – функция указатель на элемент ввода серийного номера устройства;
* getLN\_KeyAes – функция указатель на элемент ввода ключа шифрования устройства;
* getLN\_m\_settings – функция указатель на элемент ввода идентификатора устройства;
* getLN\_IPStation – функция указатель на элемент ввода IP адреса устройства;
* getBtn\_b\_setSN – функция указатель на кнопку установки серийного номера устройства;
* getBtn\_b\_GetAES – функция указатель на кнопку чтения ключа шифрования из устройства;
* getBtn\_b\_SetAES – функция указатель на кнопку установки ключа шифрования устройства;
* getBtn\_b\_send– функция указатель на кнопку передачи конфигурации в устройство;
* getBtn\_b\_rcv – функция указатель на кнопку чтения конфигурации из устройства;
* setStateTestingMode – функция установки режима unit-тестирования класса;
* isTestResult – функция чтения результата unit-тестирования класса;
* getTestMsg – функция чтения тестового сообщения при unit-тестировании класса.

4.7 FmWareDialog

Класс предназначен для осуществления процесса обновления ПО устройства. Класс содержит следующие функции:

* sigCancel – функция сигнал отмены процесса чтения/записи конфигурационных данных в устройство;
* setTitle – установка текста заголовка формы диалога;
* setMode - устанавливает режим соединения с устройством (RS232 или Ethernet);
* setRS232Name - устанавливает имя последовательного порта для связи с устройством (вызывается в случае установки режима связи – RS232);
* setSelfIP – устанавливает значение IP адреса ПЭВМ для передачи/приема данных по каналу Ethernet (вызывается в случае установки режима связи - Ethernet);
* setConfigureStr – функция установки конфигурационного плана списка устройств;
* b\_send\_clicked – функция передачи данных образа программы устройству;
* b\_rcv\_clicked – функция чтения данных образа программы из устройства;
* pdSendConfig\_cancel - функция обработчик кнопки отмены процесса чтения/записи конфигурации в устройство;
* comboBox\_TypeDev\_currentIndexChanged – обработчик выбора типа устройства;
* checkBox\_all\_stateChanged – функция выбора всех устройств для обновления ПО в списке коммуникационного плана;
* b\_loadFirmFile\_clicked – обработчик кнопки вызова диалога выбора файла образа программы;
* b\_ReadVersions\_clicked – обработчик кнопки опроса версий ПО выбранных устройств;
* closeEvent – функция обработчик сигнала закрытия окна конфигурирования;
* getIPFromCid – функция вычисления IP адреса из идентификатора устройства;
* getTypeFromCid – функция вычисления типа устройства из идентификатора устройства;
* updateListCID – функция обновления списка устройств коммуникационного плана исходя из фильтра типов устройств;
* drawCidStatus – функция отрисовки статуса проведенной операции(чтения/записи образа программы, версии);
* get\_b\_close – функция указатель на кнопку закрытия окна;
* get\_b\_loadFirmFile – функция указатель на кнопку выбора файла образа программы;
* get\_b\_ReadVersions – функция указатель на кнопку опроса версий ПО выбранных устройств;
* get\_b\_send – указатель на кнопку передачи данных образа программы устройству;
* get\_b\_rcv – указатель на кнопку чтения данных образа программы из устройства;
* get\_lineEdit\_KeyAES – указатель на элемент ввода ключа шифрования;
* get\_lineEdit\_version – указатель на элемент ввода номера версии ПО;
* get\_comboBox\_TypeDev – указатель на список типов устройств;
* get\_checkBox\_all – указатель на селектор выбора всех устройств в списке;
* setStateTestingMode – функция установки режима unit-тестирования класса;
* isTestResult – функция чтения результата unit-тестирования класса;
* getTestMsg – функция чтения тестового сообщения при unit-тестировании класса.

1. **Unit-тестирование ПО «Конфигуратор»**

Unit-тестирование представляет собой совокупность классов и методов для тестирования исправного состояния внутренних модулей программы конфигуратора. Запуск процедуры unit-тестирования инициируется запуском программы с опцией командной строки «--test». Каждый из классов unit-тестирования проверяет работоспособность или целостность интерфейса пользователя внутренних модулей программы и выдает в консоль программы сообщение об исправности или отказе внутреннего модуля программы. Тестирование выполняется путем создания экземпляра тестируемого класса и последовательного выполнения доступных функций класса с анализом выданного результата.

Результат unit-тестирования отображается в консольном окне программы. Пример отображения отрицательного результата unit-тестирования приведен на рисунке 5.1.

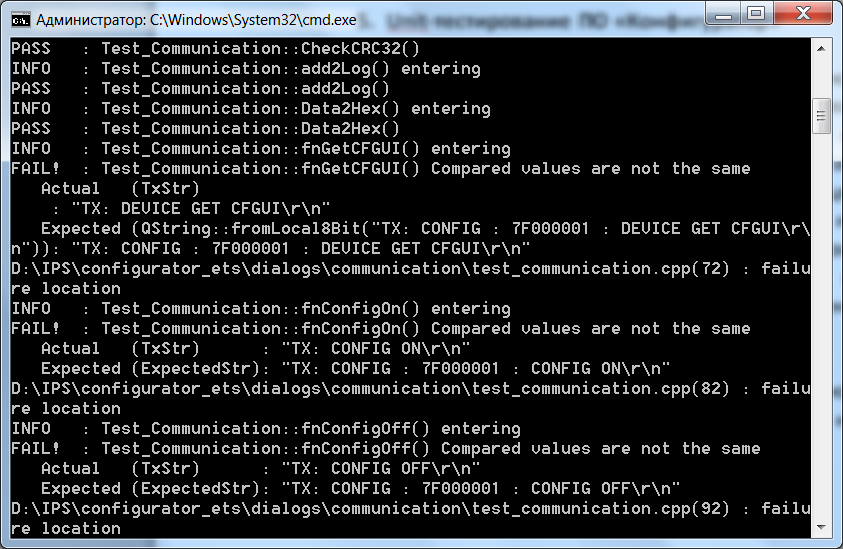


Рисунок 5.1 Консольное окно программы с отрицательным результатом unit-тестирования

Ошибки при тестировании модуля отображаются значением “FAIL” с указанием номера строки в модуле unit-теста. Положительный результат unit-тестирования подтверждается строками “PASS “ с указанием тестируемой функции модуля.

Пример отображения положительного результата unit-тестирования приведен на рисунке 5.2.

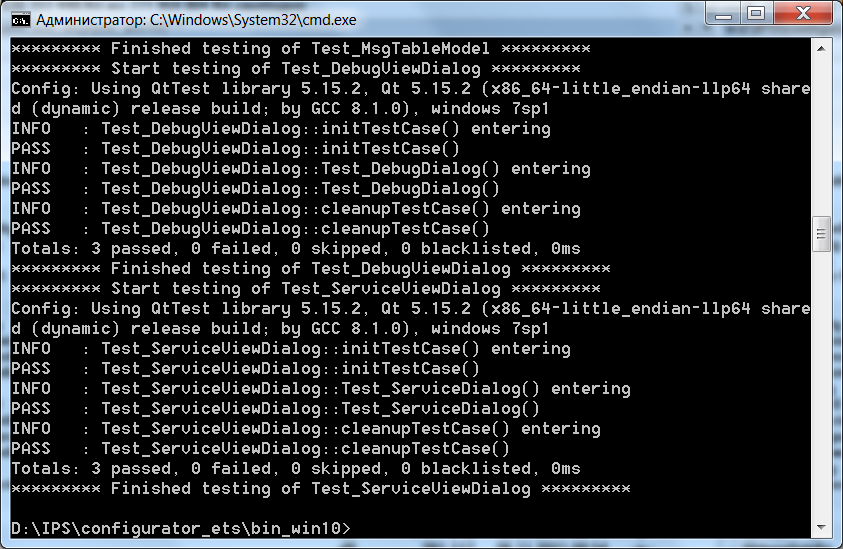


Рисунок 5.2 Консольное окно программы с положительным результатом unit-тестирования.

Список классов unit-тестирования:

* Test\_Communication - тестирование функций класса CommunicationDialog;
* Test\_CryptoDialog - тестирование функций класса CryptoDialog;
* Test\_ConfigureDialog - тестирование функций класса ConfigureDialog;
* Test\_FmwareDialog - тестирование функций класса FmwareDialog;
* Test\_DebugViewDialog - тестированиt функций класса DebugViewDialog;
* Test\_ServiceViewDialog - тестирование функций класса ServiceViewDialog.