Программа для работы с ФН

FNTest, версия 1.2

Руководство по эксплуатации

1. **Общие сведения**

1.1. Назначение и функции программы, технические средства и программное обеспечение.

Данная программа предназначена для моделирования совместной работы ККМ и ФН, тестирования ФН, демонстрации его функций, просмотра накопленных устройством данных, подготовки и печати отчетов.

Требования к рабочему месту:

* IBM PC совместимый персональный компьютер.
* Операционная система Windows7, Windows 8, Windows 10
* Коммуникационный адаптер для связи с ФН на базе микросхемы FTDI или адаптер USB-UART

Программа связывается с ФН через коммуникационный адаптер, по одному из двух интерфейсов: последовательный интерфейс UART, последовательный интерфейс I2C. В зависимости от возможностей адаптера.

1.2. Установка программного обеспечения.

Программное обеспечение поставляется в виде. Программы установки нет. Запускать на выполнение следует файл FNTest.exe.

Краткое описание назначения файлов: borlndmm.dll, cc32250mt.dll, rtl250.bpl, vcl250.bpl, vclx250.bpl – файлы поддержки системы Embarcadero С++ Builder, v 10.2. ComFun.dll, FNCmd.dll, FNPrtcl.dll, FNError.dll – поддерживающие DLL. FN\_install\_12.xml – файл с описанием команд и типов данных ФН для ФФД 1.2.

**Работа с программой**

2.1. Общие сведения

После запуска на выполнение программы FNTest.exe, на экране появляется главное окно приложения, которое разделено на три основные области (Рис.1). В верхней части окна находится главное меню, с помощью которого можно реализовать функции программы. В левой части находится список команд ФН, построенный в виде иерархически упорядоченного дерева. В правой части находится панель с закладками, на которых отображаются параметры текущей команды и результат ее выполнения.

В нижней части окна располагается статусная строка. Сразу после запуска программы выполняется первоначальное обращение к ФН для запроса статуса. Если не удалось осуществить физическую связь с устройством, на экране появиться сообщение об ошибке. В случае успешного соединения, информация о состоянии устройства помещается в статусную строку. Если необходимо исключить обращение к устройству при первоначальном запуске программы, следует запускать ее с параметром командной строки **nocheck**.



Рис.1

* 1. Главное меню

Функции программы реализуется через пункты главного меню, в некоторых ситуациях они дублируются другими элементами интерфейса, при разборе главного меню такие ситуации будут описаны отдельно.

Главное меню содержит следующие пункты:

1. Выход
2. Выполнить
3. Сервис
4. Помощь

Выбор пункта **"Выход"** позволяет закончить работу с приложением. Перед этим у пользователя будет запрошено подтверждение: "Вы действительно хотите закончить работу". При нажатии кнопки "Да"("Yes") приложение закрывается.

Выбор пункта **"Выполнить"** позволяет выполнить какую-либо команду ФН. Для той же цели можно использовать комбинацию клавиш Ctrl+Enter, кнопку [>] на панели инструментов или двойной щелчок на выбранной команде в списке команд.

Пункт **"Сервис"** содержит подпункты:

1. Параметры
2. История перерегистраций ФН
3. Сценарии тестирования (Выполнение)
4. Работа с сервером ОФД
5. Работа с ИСМ
6. Чтение фискальных документов в TLV формате
7. Очистка ФН

При выборе пункта **"Параметры"** появляется диалог для установки параметров программы.

При выборе пункта **"История перерегистраций"** в панели “Результат” выдается отчет, содержащий информацию о регистрации и всех перерегистрациях ФН.

При выборе пункта **"Работа с сервером ОФД"** появляется окно, в котором реализована функциональность, с помощью которой можно вычитывать из ФН сообщения для ОФД, просматривать их отправлять на сервер ОФД для подтверждения и анализировать ответ сервера.

При выборе пункта **"Работа с ИСМ "** появляется окно, в котором реализована функциональность, с помощью которой можно выполнять проверку КМь вычитывать из ФН сообщения для сервера ИСМ, просматривать их отправлять на сервер.

При выборе пункта **"Чтение фискальных документов в TLV формате"** появляется окно, в котором реализована функциональность, с помощью которой можно вычитывать из TLVобласти архива ФН содержащиеся там документы и просматривать их.

При выборе пункта **"Очистка ФН"** выполняется набор команд, выполняющих полную очистку отладочной ФН.

Пункт **"Помощь"** содержит единственный подпункты – “О программе”

При его выборе выводится информация о наименовании и версии данного программного продукта.

* 1. Набор команд ФН

Набор всех команд ФН представлен в виде дерева. Если элемент списка содержит дочерние элементы, то против его наименования отображается крестик. Все команды объединены в группы для удобства работы. Подробнее о командах см. в документации на ФН. Некоторые важные недокументированные отладочные команды описаны в Приложении 2 данного документа.

Перемещение по списку можно осуществлять с помощью мыши или клавиатуры, используя клавиши “Стрелка вверх” и “Стрелка вниз” для перехода к элементу списка, и клавиши “Стрелка влево” и “Стрелка вправо” для закрытия и открытия текущего уровня. При выборе какой-либо команды из списка, в правой части окна на вкладке **"Параметры"** отображается набор полей ввода, соответствующий параметрам команды. Для того чтобы перейти к вводу значений, используйте мышь или клавишу “Стрелка вправо”, а для дальнейшего перемещения по полям, любую из клавиш: "Enter", "Стрелка вверх", "Стрелка вниз", ''Tab". Для возврата в список команд можно использовать комбинацию клавиш “Ctrl+Home”. Набор команд описан в настроечном файле формата XML, который программа ищет в своей рабочей папке. Если файл найден, идентификатор набора команд добавляется в отдельный список. Допускается иметь несколько файлов с описанием команд, соответствующих различным версиям ФФД, тогда список будет содержать несколько элементов, соответствующих различным версиям ФФД. Пользователь может сам переключаться между ними, или нажать кнопку “Проверить”. В последнем случае программа сама определит версию ФФД подключенной ФН, и в случае если это возможно переключится на нужный набор команд. Программа поставляется с файлом XML для версии 1.2.

* 1. Выполнение команд ФН

Перед тем как выполнить команду, пользователь должен ввести значения параметров в соответствующие поля, при этом программа будет контролировать правильность ввода данных.

В любой момент пользователь может просмотреть содержимое протокольного блока команды на вкладке "Буфер команды" (Рис.2). Протокольный блок отображается в виде таблицы, где для каждого байта показывается:

1. Индекс байта (начиная с нуля)
2. Наименование байта (для 3-х первых и последнего байта) или наименование поля (для первого байта поля)
3. Символьное значение байта
4. Шестнадцатеричное значение
5. Десятичное значение

Находясь на вкладке "Буфер команды" пользователь может не только просматривать протокольный блок, но и редактировать его. Имеется возможность

1. Изменить значение байта
2. Вставить или удалить байт
3. Пересчитать контрольную сумму
4. Отменить все сделанные изменения

Для изменения значения байта следует просто перейти к нужной ячейке таблицы и ввести с клавиатуры новое значение. Все остальные возможности реализуются через контекстное меню, вызываемое по нажатию правой кнопки мыши. Добавление нового байта происходит перед текущим, при этом вставляется нулевой байт. Байт контрольной суммы будет записан в текущей позиции. Следует помнить о том, что при добавлении или удалении строк в таблицу будет нарушено соответствие значений и комментариев к ним.



Рис.2

После того, как параметры введены, пользователь может выполнить команду, выбрав соответствующий пункт меню или нажав клавиши Ctrl+Alt. Если команда успешно выполнена ФН и получен ответ, пользователь может просмотреть содержимое протокольного блока ответа на вкладке "Выходной буфер" или форматированный ответ на вкладке "Результат". Если в результате обработки команды произошла ошибка, на экране появится соответствующее сообщение. Подробнее о символьных сообщениях об ошибках и их кодах можно узнать в Приложении 1.

Для полноценной работы с ФН

Подавая отдельные команды ФН в определенной последовательности можно полноценно работать с ФН и использовать все его документированную функциональность, но это трудоемко, потому что практически любая операция, имеющая практическую ценность – это набор нескольких команд ФН

Для повышения удобства работы, для основных операций: регистрация, открытие и закрытие смены, оформление чека (обычного или с маркированными товарами) описаны в отдельных файлах с расширением \*.prg. Эти файлы должны находится в папке, имя которой указано в настройках приложения в поле “Папка скриптов”. По умолчанию, это рабочая папка. Программа поставляется с пятью файлами скриптов. Имя каждого файла скрипта отображается как подпункты пункта главного меню программы “Сервис \ Сценарии тестирования (выполнение)”. При выборе такого пункта немедленно начинается выполнение сценария. Описание скриптового языка содержится в приложении 3.

* 1. Настройка параметров

Параметры, определяющие работу приложения, сохраняются в системном реестре Windows. Их можно редактировать с помощью диалога "Настройка параметров", вызываемого с помощью пункта главного меню "Сервис"/"Параметры".

Диалоговое окно для настройки параметров (Рис. 3) содержит три вкладки:

1. "Общие"
2. "I2C"
3. "RS232"

На вкладке "Общие" задаются общие параметры работы программы:

1. Шрифт, с помощью которого будет отображаться текстовые данные в окнах программы
2. Тип интерфейса, по которому программа взаимодействует с ФН
3. Таймауты чтения и записи данных при обмене с ФН.
4. Папка, где хранятся скрипты команд ФН.
5. Признак "Проверять поля TLV" - если он установлен, то всякий раз при получении ответа от ФН будет проверятся корректность и наличие обязательных TLVполей документа.
6. Признак "Замерять время"-если он установлен, в логе обмена будет содержаться время выполнения каждой команды
7. Признак "Форматировать дампы" - если он установлен, то вседвоичные массивы, которые получены от ФН в качестве параметров ответа будут представлены в форматированном виде.
8. Признак "Не блокировать порт " - если он установлен, то при работе с ФН по протоколу RS232, соответствующий порт будет открываться только на время обмена с ФН. Это позволяет не блокировать порт на все время работы программы.
9. Группа полей “ОФД”, позволяет задать параметры соединения с сервером ОФД: IP-адрес, номер порта, заголовок сообщения, таймаут. Радиокнопка “Демо режим” выбирается, если подключения с сервером нет, и для тестирования ФН используется имитация подтверждения документа.
10. Группа полей “ИКМ”, позволяет задать аналогичные параметры соединения с сервером ИКМ.



Рис.3

На вкладке "I2C" задаются параметры интерфейса I2C:

1. Тип адаптера (программа поддерживает 4 варианта).
2. Дополнительная информация

Основным способом подключения по интерфейсу I2c является использование адаптера FTDI C232HM. Серийный номер указывается, только если к компьютеру подключено несколько адаптеров данного типа.

На вкладке "RS232" задаются параметры интерфейса RS232:

1. Номер COM порта.
2. Скорость COM порта.
	1. Работа с сервером ОФД

При выборе пункта главное меню “Работа с сервером ОФД” отображается окно, содержащее панель быстрых кнопок с помощью которого можно реализовать функции взаимодействия с ОФД и две панели для отображения результатов такого взаимодействия (Рис.4). В левой панели содержатся данные обмена с ОФД в HEX виде, а в правой те же данные, но в текстовом, разобранном (где это возможно) виде.

Назначение кнопок кнопочного меню (описание будет дано слева направо):

Кнопка 1 – позволяет открыть предварительно сохраненное в файле сообщение для ОФД

Кнопка 2 – позволяет сохранить в файле сообщение.

Кнопка 3 – очищает панели данных.

Кнопка 4 – получает из ФН очередное сообщение для ОФД (выполняется команды 0x24, 0x21, 0x22, 0x23, 0x25 ФН для полной вычитки сообщения)

Кнопка 5 – отправляет, вычитанное из ФН сообщение на сервер ОФД, получает ответ и выполняет команду ФН 0x26.

Кнопка 6 – выполняет получение сообщения из ФН и отправку его на сервер.

Кнопка 7 – выполняет проверку имитовставки в принятой от сервера ОФД квитанции.

Кнопка 8 – выполняет получение, подтверждение на сервере ОФД и сохранение в файлах всех сообщений для ОФД, которые имеются в ФН.

Поле выбора “Очищать лог при каждом вызове” позволяет очищать или напротив накапливать информацию при каждом вызове.



Рис.4

* 1. Работа с ИСМ

При выборе пункта главное меню “Работа с ИСМ ” отображается окно, содержащее панель быстрых кнопок с помощью которого можно реализовать функции взаимодействия с информационным сервером маркировки две панели для отображения результатов такого взаимодействия (Рис.5). В левой панели содержатся данные обмена с ИСМ в HEX виде, а в правой те же данные но в текстовом, разобранном (где это возможно) виде.

Назначение кнопок кнопочного меню (описание будет дано слева направо):

Кнопка 1 – очищает панели данных.

Кнопка 2 – Выполняет проверку кода маркировки на ИСМ. ВАЖНО: перед выполнением данной функции следует выполнить команды ФН 0xB1 и 0xB5, в главном окне программы. Эти команды сформируют буфер для проверки, который и будет отправлен на ИСМ по нажатию этой кнопки. Результат обмена с сервером будет отображен на информационных панелях (см. рис. 5). Кнопка 3 – вычитывает из ФН очередное уведомление о реализации маркированного товара и помещает его во внутренний буфер программы. Для вычитывания уведомления используются команды ФН 0xBB, 0xBC, 0xBE. Содержимое уведомления будет показано на информационных панелях.

Кнопка 4 – выполняет отправку предварительно считанного уведомления на сервер ИСМ, после получения ответа от сервера он передается ФН с помощью команды 0xBF. Ответ сервера и ФН отображаются на информационных панелях.

Поле выбора “Очищать лог при каждом вызове” позволяет очищать или напротив накапливать информацию при каждом вызове.



Рис.5

* 1. Чтение фискальных документов в TLV формате

При выборе пункта главное меню “Работа Чтение фискальных документов в TLV формате” отображается окно, позволяющее просматривать фискальные документы хранящиеся в области TLV документов ФН. Окно содержит поле ввода “Номер ФД“ для указания номера документа, трех кнопок управления и двух панели для отображения документов (Рис.6). В левой панели содержатся данные в HEX виде, а в правой те же данные, но в текстовом, разобранном (где это возможно) виде.

Кнопка 1 – извлекает из ФН документ номер которого указано в поле “Номер ФД“

Кнопка 2 – извлекает следующий документ и инкрементирует номер в поле “Номер ФД“

Кнопка 3 – извлекает предыдущий документ

Поле выбора “Очищать лог при каждом вызове” позволяет очищать или напротив накапливать информацию при каждом вызове.



Рис.6

**Приложение 1**

Сообщения об ошибках.

В приложении приведены сообщения об ошибках, генерируемые программой, их коды и наиболее вероятные причины их возникновения.

1) Ошибки коммуникационных портов (COM, LPT)

0x101 - "Не открывается коммуникационный порт"

 Возможно порт PC неисправен

0х102 - "Коммуникационный порт уже открыт"

 Проверьте, не запущено ли другое приложение, которое пользуется теми же самыми коммуникационными ресурсами, что и FNTest.

0x103 - "Неверно инициализирован"

 Возможно устаревшая микросхема порта не поддерживает установленную Вами скорость обмена данными.Откорректируйте параметры программы.

0x104 - "Истек таймаут"

 Данное сообщение появляется всякий раз, когда операции чтения или записи не были закончены в установленный временной интервал. Откорректируйте параметры программы.

0x105 - "Не удалось найти vicx95.dll"

 Поместите файл динамической библиотеки в рабочий каталог программы FNTest.exe или определите путь поиска для файла vicx95.dll (данная библиотека используется для доступа к LPT порту в среде Windows и используется только в случае когда протокол I2Cэмулируется с помощью LPT порта PC).

2)Ошибки протокола

0x201 - "Ошибка записи в порт"

Не удалось записать в коммуникационный порт требуемого количества байт.

0x202 - "Ошибка чтения из порта"

Не удалось прочитать из коммуникационного порта требуемого количества байт.

0x203 - "Нарушение протокола"

Данное сообщение появляется всякий раз, когда нарушается порядок следования данных, предписанный протоколом обмена между PC и ФН.

0x206 - "Неверная контрольная сумма"

 Данные, от ФН были искажены при передаче.

3)Ошибки устройства

 Все коды ошибок данной группы, соответствуют значениям статусного байта ответа ФН на команду (см. документацию на ФН).

0x301 - Некорректный формат или параметр команды

0x302 - Неверное состояние ФН

0x303 - Ошибка ФН

0x304 - Ошибка КС

0x305 - Закончен срок эксплуатации ФН

0x306 - Архив ФН переполнен

0x307 - Неверные дата и/или время

0x308 - Нет запрошенных данных

0x309 - Некорректное значение параметров команды

0x30A - Неверная команда

0x30B - Этот тег формирует ФН 1.1.

0x30C- Дублирование тега

0x30D - Отсутствуют данные, необходимые для корректного учета в ФН

0x30E - Количество позиций, подлежащих учету в документе, превысило разрешенный лимит

0x30F - Отсутствуют данные в команде

0x310 - Превышение размеров TLV данных

0x311 - Нет транспортного соединения

0x312 - Исчерпан ресурс КС

0x314 - Исчерпан ресурс ФН

0x315 - Исчерпан ресурс ожидания передачи сообщения

0x316 - Продолжительность смены не может быть более 24 часов

0x317 - Разница между операциями не совпадает с таймером

0x318 - Содержание тега имеет недопустимый формат

0x319 - Продажа подакцизного товара

0x31A - Неустановленный реквизит в товарной позиции

0x320 - Сообщение от ОФД не может быть принято

0x332 - Запрещена работа с маркированными товарами

0x333 - Неверная последовательность команд при работе с маркированными товарами

0x334 - Работа с маркированными товарами временно заблокирована

0x335 - Переполнена таблица хранения кодов маркировки

0x33С - В блоке TLV отсутствуют необходимые реквизиты

0x33E - Тэг 2007 содержит не проверенный ранее реквизит

0x340 - Неверный ПИН

**Приложение 2**

Отладочные команды ФН, которые могут быть использованы для тестировании только отладочных версий ФН.

1. Команда “Сброс состояния ФН”.

 Код команды 0х60.

 Входные параметры:

1. 1 байт – Тип операции (значения 24, 25, 22, 33)

 Выходные параметры:

1. 1 байт – Результат операции

Выполнение команды:Взависимости от типа операции ФН выполняет следующие действия:

22 – Очистка флагов ФН

24 - Начало очистки всей флэш памяти ФН (операция может занять десятки секунд времени)

25 – Запрос состояния очистки память

33 – активизация встроенного бутлоадера для перезагрузки ПО на отладочных ФН

**Приложение 3**

Описания скриптового языка.

В приложении приведено краткое описание макроязыка, применяемого при написании сценариев тестирования ФН. В тексте использованы следующие обозначения: символы "[ ... ]" - задают необязательные значения, символы "{ ... }" - обозначают список значений, символ "/" разделяет альтернативные значения.

Текст сценария может содержать:

1. Переменные
2. Константы
3. Операторы цикла
4. Условный оператор IF
5. Логические выражения
6. Оператор вывода сообщения
7. Оператор присваивания
8. Макрокоманды
9. Комментарии
10. Системные константы
11. Оператор DEFINE
12. Оператор выхода из программы EXIT
13. Оператор останова программы
14. **Переменные**

Переменные, если они используются в сценарии, должны быть обязательно описаны. Описание переменных должно предшествовать любым командам или операциям с ними. Секция, в которой описываются переменные, начинается с ключевого слова **VAR** и заканчивается словом **ENDVAR**. Эта секция должна быть единственной, количество переменных не ограничено. Переменные могут быть проинициализированы в месте их описания. Все переменные глобальные, т.е. область видимости любой переменной - весь сценарий.

Синтаксис:

 **Имя\_типаИмя\_переменной[ = начальное значение переменной ]**

**Имя типа**

Всего в макроязыке существует пять типов данных:

1) **INTEGER -** обобщенный знаковый 8 байтовый целый тип. Целые константы могут быть записаны в шестнадцатеричном виде, для этого их снабжают префиксом "0x".

2) **DOUBLE -** обобщенный 8 байтовый вещественный тип. При записи констант данного типа используется десятичный разделитель, выбранный в региональных настройках Windows.

3) **STRING -** строка

4) **DATE** - тип "дата" (формат записи констант определяется региональными настройками Windows - например 12.04.2001).

5) **TIME** - тип "время"(формат записи констант определяется региональными настройками Windows - например 12:01).

Типы данных переменных должны быть согласованы с типом данных ФН.

**Имя переменной** - Строка из не более чем 256 символов, в которой допускаются цифры, буквы латинского и русского алфавита и символ '\_'.

**"="** - оператор присваивания.

**Начальное значение переменной** задается строкой, определяемой типом данных (см. выше)

Пример описания переменных:

 **VAR**

 **. . .**

 **Datedata = 12.06.2001**

 **Time time = 19:30**

 **Sum sum**

 **SectionNumbersect\_num**

 **. . .**

 **ENDVAR**

1. **Константы**

Константы используются в правой части операции присваивания, в операторе цикла и непосредственно в тексте макрокоманд.

О способах записи констант было сказано выше.

1. **Операторы цикла**

В тексте сценария можно задавать любое количество циклов. Циклы могут быть вложенными. Степень вложенности не ограничивается. Циклы могут быть условными и безусловными. Безусловный цикл начинается с ключевого слова **DO** и заканчивается словом **ENDDO**. После слова **DO**, через пробел,должна обязательно следовать константа или переменная целого типа, задающая количество повторов,например:

 **DO 10**

 **. . .**

 **тело цикла**

 **. . .**

 **ENDDO**

Условный цикл начинается с ключевого слова **DOWHILE** и заканчивается словом **ENDDOWHILE**. После слова **DOWHILE**, через пробел,должно следовать логическое выражение задающее условие выполнения цикла. Цикл выполняется до тех пор, пока выражение истинно. О правилах записи логических выражений см. ниже. Пример условного цикла:

 **date = 05.10.01**

 **DOWHILEdate<=10.10.01**

 **. . .**

 **тело цикла**

**INC date 1 ;увеличиваем на 1 день . . .**

 **ENDDOWHILE**

1. **Условный оператор**

В тексте сценария можно задавать простые ветвления используя также условный оператор **IF,**

его синтаксис:

 **IF логическое\_выражение**

 **. . .**

 **ENDIF**

Произвольный блок кода, заключенный между словами **IF** и **ENDIF** будет выполняться если логическое выражение истинно. О правилах записи логических выражений см. ниже.

**5Логические выражения**

Логические выражения используются в условных операторах. Они могут содержать имена переменных, константы (в том числе системные) и один из логических операторов: **”<”** – меньше, **”>”** - больше!, **”<=”** - меньше или равно, **”>=”** - больше или равно, **”==” -**  равно, **”!=” -**  не равно, **”**&&**”** – логическое "И", **”**||**”** – логическое "Или".

Например, выражение **date>=10.10.00 &&date<=10.10.01** истинно, если значение переменной **data**принадлежит отрезку времени от десятого октября 2000 года до десятого октября 2001.

1. **Оператор вывода сообщения**

Для вывода текстового сообщения на экран монитора, предназначена команда **SHOWMESSAGE.**

Ее синтаксис: **SHOWMESSAGE Переменная / Текстовая константа.** Где переменная должна иметь тип **STRING**.

1. **Оператор присваивания**

Оператор присваивания задается символом "=". Он необходим для выполнения операции присваивания, которая записывается в виде: **Имя\_переменной =Имя\_переменной / Константа**. Переменная к этому моменту уже должна быть описана, правила записи констант приведены выше.

Пример: **sect\_num = 1**

1. **Макрокоманды**

В сценарии могут быть использованы макрокоманды: **COMMAND, ANSWER, RAND\_SET, INC, DEC**

Разделителем параметров макрокоманд всюду является пробел. Переносы строк при записи макрокоманд не допускаются.

Макрокоманда **COMMAND** позволяет выполнить произвольную команду ФН, ее синтаксис:

 **COMMAND Код команды [{Список параметров команды}]**

**Код команды -** Константа, задающая код команды

**{Список параметров команды}** - список констант или имен переменных, задающих параметры команды.

Разделителем всюду является пробел.

Макрокоманда **ANSWER** позволяет сохранить в переменных параметры ответа на последнюю команду, ее синтаксис:

 **ANSWER {Список переменных}**

**{Список переменных}** - список имен переменных, в которых будут сохранены результаты. Тип переменных **обязательно** должен соответствовать типу параметров ответа (см. выше). Разделителем всюду является пробел.

Макрокоманда **RAND\_SET** позволяет присвоить переменной случайное значение из списка, заданного в параметрах команды, ее синтаксис:

 **RAND\_SET Имя переменной {список значений}**

**Имя переменной -** Имя переменной, описанной ранее.

**{список значений}** - Список констант, разделенных символом ';'. Длина списка не ограничена.

Макрокоманда **INC** позволяет увеличить значение переменной на величину указанной константы или переменной:

 **INC Имя\_переменной/Константа Имя\_переменной/Константа**

Макрокоманда **DEC** позволяет уменьшить значение переменной на величину указанной константы или переменной:

 **DEC Имя\_переменной/Константа Имя\_переменной/Константа**

**Замечание:** Описанные команды **не могут быть** выполнены для переменных типа **STRING**, команда **DEC не может быть выполнена** для переменных типа **DATE** и **TIME**, а при выполнении команды **INC,**для переменной типа**DATE,** вторым операндом должно быть количество **дней**, a для переменной типа **TIME** количество **минут**.

1. **Комментарии**

Любая строка в тексте сценария, начинающаяся с символа ";" считается комментарием и интерпретатором не обрабатывается.

1. **Системные константы**

При записи операции присваивания или команды **COMMAND** можно использовать именованные константы: **$RetCode**(значение статусного байта в ответе ФН на последнюю команду), **$CurDate** (текущая системная дата), **$CurTime**(текущее системное время), **$LastPage** (страница архива с которой начинаются записи последней смены - эту величину можно использовать для того, чтобы оценить заполненность архива).

1. **Оператор DEFINE**

Этот оператор должен стоять в начале программы. Он предназначен для присвоения значений системным переменным $**LogName,$ErrLogName, $MessageLogName,$ShowFnError, $FormatLog**. Если значения переменным $**LogName,$ErrLogName, $MessageLogName**не присвоены, никакие протоколы при выполнении сценария вестись не будут. Если переменной $**LogName**присвоено строковое значение, оно воспринимается как имя файла где будет вестись протокол выполнения. Значение переменной **$FormatLog**при этом определяет будет ли протокол выводиться в форматированном (**$FormatLog=Y**) или неформатированном (**$FormatLog=N**) виде. В переменной **$ErrLogName**задается имя файла протокола ошибок выполнения. В файл сообщений программы, имя которого задается в переменной **$MessageLogName**будут помещены строки всех сообщений, выводимых в сценарии с помощью команды **SHOWMESSAGE** (по умолчания эти сообщения выводятся на экран PC). Использования файлов протокола уменьшает время выполнения, экономит оперативную память и позволяет выполнить сценарий в полностью автоматическом режиме.

 Для того чтобы подавить вывод сообщений об ошибках ФН (см. Приложение 1, пункт 3 - "Ошибки устройства"), при выполнении сценария тестирования, необходимо присвоить переменной **$ShowFnError**значение 0.

Пример: **DEFINE $LogName C:\TEMP\LOGFILE.TXT**

 **DEFINE $FormatLogY**

1. **Оператор EXIT**

Предназначен для прекращения выполнения программы. Может стоять в любом месте программы кроме блока описания переменных.

1. **Оператор останова**

Для временного останова выполнения программы следует установить в отдельной строке символ **"#"**.

Выполнение программы будет приостановлено именно на этой строке. Дальнейшее выполнение возможно в пошаговом или непрерывном режиме. Для этого надо воспользоваться пунктами меню **“Выполнить”**или**“Шаг”**соответственно (или кнопками на панели управления).