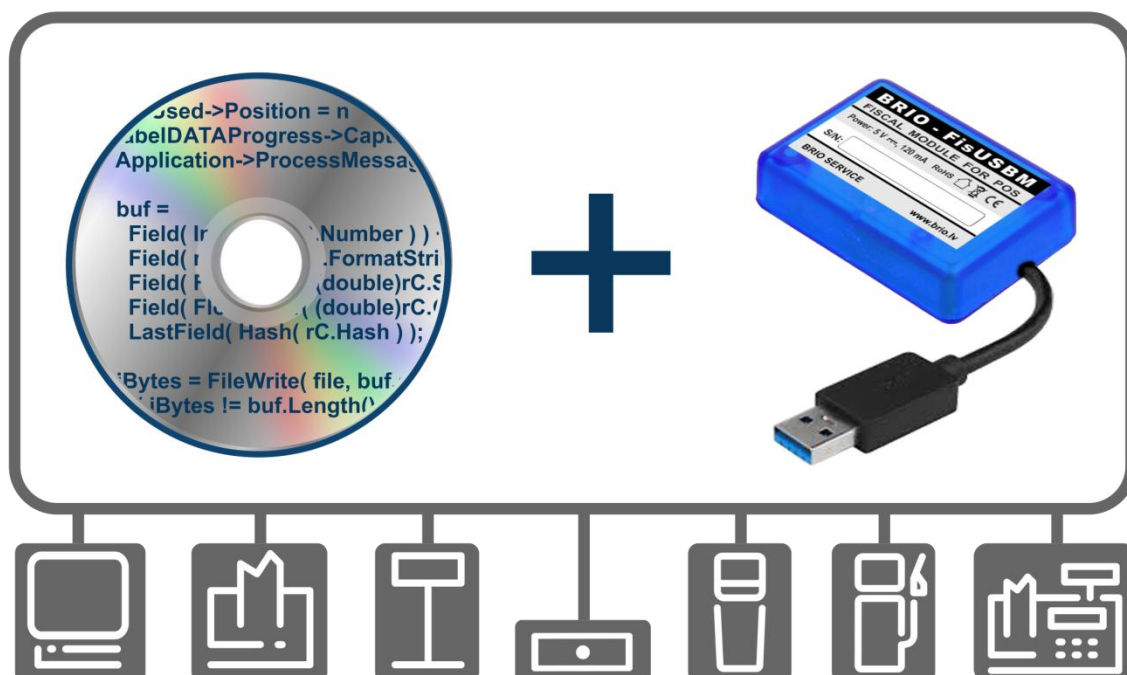


BRIO FisUSBM



FISKĀLA RISINĀJUMS POS-iem.

Редакция от 19.12.2016

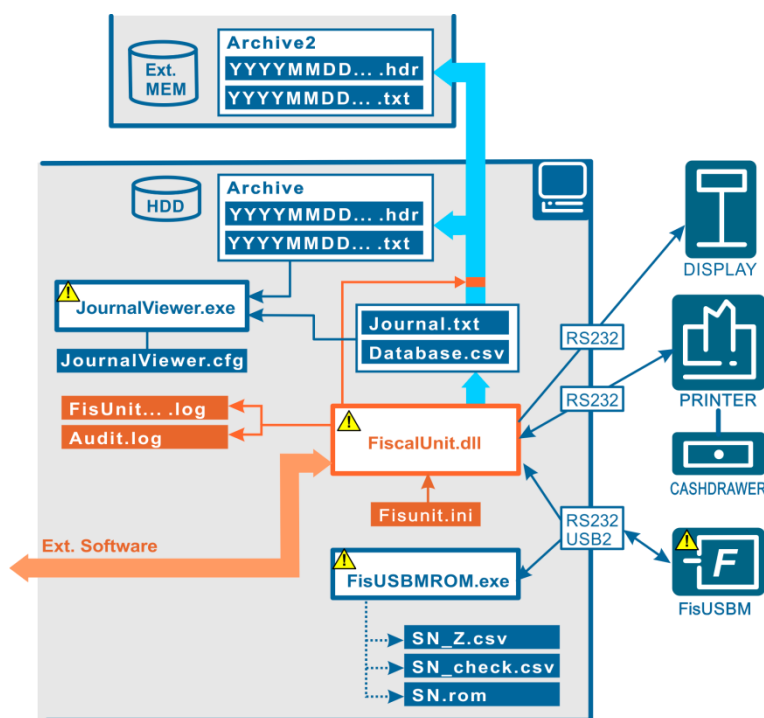
| | |
|---|-----------|
| 1. ФИСКАЛЬНОЕ РЕШЕНИЕ FisUSBM | 3 |
| 1.1. НАЗНАЧЕНИЕ | 3 |
| 1.2. СТРУКТУРА И СОСТАВ ФИСКАЛЬНОГО РЕШЕНИЯ | 3 |
| 2. ФИСКАЛЬНЫЙ МОДУЛЬ FisUSBM | 6 |
| 2.1. ПАРАМЕТРЫ..... | 6 |
| 2.2. ТИПЫ КОРПУСОВ ФМ | 7 |
| 2.3. СИГНАЛЫ РАЗЪЕМОВ ИНТЕРФЕЙСОВ | 7 |
| 2.4. ИНСТАЛЛЯЦИЯ ФИСКАЛЬНОГО МОДУЛЯ..... | 8 |
| 2.4.1. ФМ С ИНТЕРФЕЙСОМ USB..... | 8 |
| 2.4.2. ФМ С ИНТЕРФЕЙСОМ RS-232 | 9 |
| 3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ФР FisUSBM | 10 |
| 3.1. РАЗДЕЛЫ И ФАЙЛЫ | 10 |
| 3.2. ФАЙЛ ИНИЦИАЛИЗАЦИИ FISUNIT.INI | 10 |
| 3.2.1. ПРИМЕРЫ ФАЙЛОВ ПРОТОКОЛОВ И АУДИТА | 11 |
| 3.3. ПРОГРАММА FisUSBROM.EXE | 12 |
| 3.3.1. НАЗНАЧЕНИЕ ПРОГРАММЫ | 12 |
| 3.4. ПРОГРАММА JOURNAL VIEWER.EXE | 12 |
| 3.4.1. НАЗНАЧЕНИЕ И НАСТРОЙКИ | 12 |
| 3.4.2. ГЛАВНОЕ ОКНО ПРОГРАММЫ | 12 |
| 3.4.3. ПРОСМОТР И ПРОВЕРКА ЛЕНТЫ | 13 |
| 3.4.4. РАСЧЕТ ЗНАЧЕНИЯ HASH SHA-1 ДЛЯ ФРАГМЕНТА ЛЕНТЫ | 14 |
| 3.4.5. ПРИМЕРЫ ФРАГМЕНТОВ ФАЙЛОВ..... | 15 |
| 4. ПРИЛОЖЕНИЯ | 16 |
| 4.1. БИБЛИОТЕКА ФУНКЦИЙ FiscalUNIT. | 16 |
| 4.1.1. УСТАНОВОЧНЫЕ ФУНКЦИИ | 16 |
| 4.1.2. ИНФОРМАЦИОННЫЕ ФУНКЦИИ | 19 |
| 4.1.3. СИСТЕМНЫЕ ФУНКЦИИ | 20 |
| 4.1.4. ФУНКЦИИ ФИСКАЛЬНЫХ ЧЕКОВ | 21 |
| 4.1.5. ФУНКЦИИ НЕ ФИСКАЛЬНЫХ ЧЕКОВ | 24 |
| 4.1.6. СЕРВИСНЫЕ ФУНКЦИИ | 26 |
| 4.1.7. АТТРИБУТЫ ПЕЧАТИ | 27 |
| 4.1.8. КОДЫ ЗАПРАШИВАЕМОЙ ИНФОРМАЦИИ | 27 |
| 4.1.9. ТИПЫ ПРИНТЕРОВ | 27 |
| 4.1.10. ТИПЫ ПОЛЬЗОВАТЕЛЕЙ..... | 27 |
| 4.1.11. КОДЫ СОСТОЯНИЯ МОДУЛЯ | 28 |
| 4.1.12. ТИПЫ ШТРИХ-КОДОВ | 28 |
| 4.2. КОДЫ ОШИБОК..... | 29 |
| 4.2.1. ОШИБКИ ОТ FiscalUnit.DLL | 29 |
| 4.2.1. ОШИБКИ, ТРАНСЛИРУЕМЫЕ ОТ FisUSBM..... | 29 |

1. ФИСКАЛЬНОЕ РЕШЕНИЕ FisUSBM

1.1. НАЗНАЧЕНИЕ

- Фискальное Решение FisUSBM (ФР FisUSBM) предназначено для использования в POS системах и соответствует правилам [Министру Kabineta noteikumu Nr. 95 un Nr. 96](#)
- ФР FisUSBM состоит из нового Фискального Модуля FisUSBM (ФМ) и набора программного обеспечения.
- Программное обеспечение позволяет легко интегрировать ФР в компьютерные системы сторонних производителей.
- Фискальный Модуль, входящий в состав Фискального Решения, выпускается в трех типах корпусов, с одним из двух типов интерфейсов RS232 или USB2.
- ФМ, имеющий одновременно оба интерфейса (RS232 и USB2), хоть и является полностью работоспособным, но предназначен для тестирования и разработки ПО сторонними производителями.
- Набор команд и библиотеки доступных функций ФР, за исключением нескольких изменений, совместимы с командами Фискального Модуля BRIO-FM32E.
- Каждый ФМ имеет **уникальный серийный номер**, который присваивается производителем микросхем, используемых в ФМ.

1.2. СТРУКТУРА И СОСТАВ ФИСКАЛЬНОГО РЕШЕНИЯ



ВНИМАНИЕ!!! Hash-код программ, помеченных значком , предполагается сдавать в СГД .

- PC – Компьютер POS-а.
 - Компьютер POS-а с установленной операционной системой Windows .
- HDD – Внутренний жесткий диск компьютера.

- Устройство памяти, расположенное в корпусе компьютера POS-а.
- **Ext. MEM – Внешнее устройство памяти.**
 - Устройство памяти, расположенное вне корпуса POS-а, на котором должна храниться вторая копия архива данных (Archive2)
- **FisUSBM – Фискальный модуль.** Предназначен для хранения фискальной информации.
 - По каждому подготовленному документу:
 - Номер, дата и время соответствующего документа,
 - Контрольная сумма документа (SHA-1)
 - Grand total,
 - Для документа вида "сделка" - сумма сделки в денежном выражении.
 - При распечатке Z отчета:
 - Номер Z отчета, дата и время распечатки,
 - Общая сумма зарегистрированных в отчетном периоде (положительных) сделок "чек"
 - Общая сумма возврата, общая сумма аннулирования, общая сумма закупки в денежном выражении.
 - Примененные ставки налога на добавленную стоимость, распределение облагаемой налогом на добавленную стоимость суммы по примененным ставкам и соответствующие налоговые суммы.
 - Рассчитанная за отчетный период контрольная сумма контрольной ленты (SHA-1) и Grand total.
 - Перед использованием кассовой системы:
 - Название, регистрационный номер или персональный код пользователя.
 - Номер шасси кассовой системы, а также дата и время данной записи.
 - Для расчета общей суммы сделок периода Z отчета зарегистрированная программа системы использует сохраненную в фискальной памяти информацию о положительных сделках конкретного отчетного периода, которые сохранены в ФМ.
- **Display – Дисплей покупателя.**
 - Подключается к компьютеру по последовательному интерфейсу ComNN (RS-232).
 - Возможно подключение любых моделей двухстрочных дисплеев, совместимых по системе команд и интерфейсу с моделями CD7220, Epson, Bixolon.
- **Printer – Принтер для печати чеков.**
 - Подключается к компьютеру по последовательному интерфейсу ComNN (RS-232).
 - Возможно подключение любых моделей принтеров, совместимых по системе команд и интерфейсу с принтерами: Epson TM-U210A, Epson TM-U220A, Epson TM-U260A, Bixolon SRP-270D, Bixolon SRP-350Plus, Partner Tech RP-300, Bixolon SRP-275C, Bixolon SRP-370, Bixolon SRP-500, Partner Tech PT-6200
- **Cashdrawer – Ящик для денег.**
 - Подключается к соответствующему порту Принтера, или POS-а.
 - Управляется посредством команд, подаваемых на чековый принтер, или интерфейс POS-а.
- **FisUSBMROM.exe** – Программа для контролирующих органов. Позволяет считывать и сохранять в отдельных файлах данные, полученные непосредственно из ФМ.
 - SerialNumber_Z.csv – Текстовый файл с копиями всех Z-отчетов, хранящихся в ФМ.
 - SerialNumber_check.csv – Текстовый файл с копиями суммарных данных по всем Чекам, хранящимся в ФМ.
 - SerialNumber.gom – Двоичный файл с копией внутренней программы ФМ (Firmware).
- **FiscalUnit.dll** – Библиотека для работы с ФМ, и периферийным оборудованием.
 - Управляет работой ФМ.
 - Управляет работой чекового принтера и ящика для денег.
 - Управляет работой дисплея покупателя.
 - Обеспечивает взаимодействие с внешними программами посредством набора функций.
 - Создает электронные контрольные ленты (Journal.txt), данные продаж (DataBase.csv)
 - Создает файлы архива (Archive) текущей контрольной ленты и данных продаж (YYYYMMDD_NNNN.hdr и YYYYMMDD_NNNN.txt) после очередного Z-отчета.

- Создает вторую копию файлов архива (Archive2) и записывает ее на внешний носитель.
- Создает файлы логирования процесса обмена с ФМ (FisUSBM_DDMMYYYY.log) и/или внешней программой (fisunit_DDMMYYYY_fisunit.log).
- Ведет файл журнала регистрации пользователей - Audit.log

- **JournalViewer.exe** – Программа просмотра электронных контрольных лент.
 - Позволяет просматривать контрольные ленты.
 - Проверяет целостность контрольных лент.
 - Осуществляет поиск текстового фрагмента в лентах по запросу пользователя.
 - Рассчитывает **Hash SHA-1** выделенного фрагмента текста контрольной ленты.

2. ФИСКАЛЬНЫЙ МОДУЛЬ FisUSBM

2.1. ПАРАМЕТРЫ

- Интерфейсы: - USB-2, RS-232
- Скорость обмена по RS-232: - 115200 Bits/sec. 8 Bits, NP, 1-Stop, Non flow control
- Режим обмена: - Пакетный обмен по специальному протоколу.
- Разъемы интерфейса RS232: - D-Sub DB-9F, FC-10P
- Разъемы интерфейса USB2: - USB-A, MicroUSB-B
- Неизменяемый серийный номер. - 64 битный уникальный, не изменяемый и не стираемый серийный номер.
- Объем внутренней памяти. - 512 MBit
- Максимальное количество Чеков. - 1'000'000
- Максимальное количество Z-Отчетов. - До 2'000
- Габаритные размеры (mm):
 - Корпус типа P: 50 x 35 x 15
 - Корпус типа S: 80 x 40 x 15
 - Корпус типа M: 60 x 35 x 19
 - Корпус типа H: 60 x 35 x 19 с планкой
- Защита внутренней программы и данных: - Реализована аппаратная защита на схемотехническом уровне от стирания и/или изменения.
- Контроль даты и времени данных: - Встроенный контроль, запрещающий производить запись данных, если их дата и время меньше, чем дата и время последних сохраненных данных.
- Контроль даты и времени Grand Total: - Встроенный контроль, запрещающий производить запись очередного Grand total, если его значение меньше, чем значение последнего записанного Grand total.
- Напряжение питания: - DC +5V, +- 10%
- Максимальный ток потребления: - 100 mA
- Условия окружающей среды:
 - Температура - - 5 °C ... + 50 °C
 - Относительная влажность - 80% (Кратковременно до 95 % при 25 °C)
 - Атмосферное давление - 86...107 кПа
- Защита от воздействия пыли и влаги. - IP54

2.2. ТИПЫ КОРПУСОВ ФМ

- Фискальный Модель может выпускается в четырех типах корпусов и с двумя типами интерфейсов.

FisUSBM – P R



Тип интерфейса

- R = RS-232C
- U = Только USB2

Тип корпуса:

- P = Пластиковый корпус 50x35x15;
- S = Пластиковый корпус 80x40x15;
- M = Металлический корпус;
- H = Металлический корпус с планкой.



FisUSBM – P – XX

- Корпус из цветного GP ABS пластика.
- Размер: 50 X 30 X 15



FisUSBM – S – XX

- Корпус из цветного GP ABS пластика.
- Размер: 80 X 40 X 15



FisUSBM – M – XX

- Корпус из металла с защитным покрытием.
- Размер: 60 X 35 X 19
- Дополнительный провод заземления.

ВНИМАНИЕ!!! Свободный конец провода заземления следует надежно соединить с металлическим корпусом устройства, максимально близко к блоку питания.

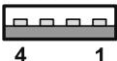



FisUSBM – H – XX

- Корпус из металла с защитным покрытием.
- Размер: 60 X 35 X 19
- Стандартная планка для установки в компьютер.

2.3. СИГНАЛЫ РАЗЪЕМОВ ИНТЕРФЕЙСОВ

| RS-232C | | FC-10P | | | |
|---------|-------|--------|-------|-----------------|--|
| | | | | | |
| Конт. | Конт. | Сигнал | | Назначение | |
| 3 | 2 | RxD | Вход | Прием данных | |
| 2 | 3 | TxD | Выход | Передача данных | |
| 6 | 4 | DTR | Выход | | |
| 5 | 5 | GND | | Общий провод | |
| 4 | 6 | DSR | Вход | | |
| 9 | 9 | DC +5V | Вход | Питания DC +5V | |

| USB-2 | | | |
|---|---|--------|----------------|
| USB-A | MicroUSB-B | | |
|  |  | | |
| Конт. | Конт. | Сигнал | Назначение |
| 1 | 1 | DC +5V | Питания DC +5V |
| 2 | 2 | D - | Данные |
| 3 | 3 | D + | Данные |
| 4 | 5 | GND | Общий провод |

2.4. ИНСТАЛЛЯЦИЯ ФИСКАЛЬНОГО МОДУЛЯ

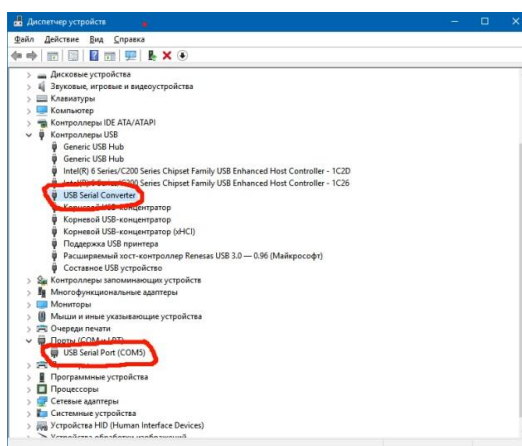
2.4.1. ФМ С ИНТЕРФЕЙСОМ USB

- Вставьте интерфейсный кабель ФМ в USB порт компьютера. На Фискальном Модуле должен загореться светодиод питания.

ВНИМАНИЕ!!! Дальнейшие действия по установке драйвера зависят от версии операционной системы Windows. Например, Windows-10 самостоятельно распознает фискальный модуль и автоматически загрузит и установит драйвера (При наличии подключения к Интернету).

ВНИМАНИЕ!!! Если ваша версия Windows определила ФМ как НЕИЗВЕСТНОЕ УСТРОЙСТВО, необходимые драйвера можно загрузить по ссылке <http://www.ftdichip.com/Drivers/D2XX.htm>.

- После того, как Windows запросит драйвера для подключенного ФМ, укажите путь к разделу, содержащему эти драйвера.
- В случае успешной установки драйверов, в Device Manager системы появятся два новых устройства:
 - USB Serial Converter
 - USB Serial Port (ComNN).



ВНИМАНИЕ!!! Номер Com-NN порта для ФМ система присваивает сама. Не изменяйте его!

2.4.2. ФМ С ИНТЕРФЕЙСОМ RS-232

- При подключении к компьютеру ФМ по интерфейсу RS-232 никакой специальной инсталляции драйверов не требуется.
- Все настройки для работы с ФМ прописываются в файле FISUNIT.INI в разделе [Port]. Укажите номер порта, к которому подключен ФМ, и скорость обмен 115200.

```
[Port]
port=com1
baud=115200
timeoutRead=5000
timeoutSend=1000
```

ВНИМАНИЕ!!! Если напряжение питания, которое ФМ должен получать с 9-го контакта разъема RS-232, превышает +5.5V - это приведет к повреждению модуля!

3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ФР FisUSBM

3.1. РАЗДЕЛЫ И ФАЙЛЫ

- Создайте на диске, на котором будет установлено программное обеспечение для ФМ, новый раздел, например – **Fisusbm**.
- Скопируйте в него с инсталляционного носителя следующие разделы и файлы:

```

.\Fisusbm\
├─ Journal\ Раздел для текущего журнал, файла данных и журнала аудита.
│  └─ Archive\ - Основной раздел архива журналов и файлов данных.
│     ├── YYYYMMDD_NNNN.hdr - Контрольный файл Электронной Ленты.
│     └── YYYYMMDD_NNNN.txt - Файл Электронной Ленты.
│  └─ FMB\ - Раздел для хранения буфера обмена с принтером.
│  └─ Audit.log - Текущий журнал аудита.
│  └─ Journal.txt - Текущая контрольная лента.
│  └─ Database.csv - Текущая база данных.
│  └─ Database.hdr - Контрольный файл Базы Данных.
│  └─ Report.db - База Данных для формирования периодических отчетов по Z-отчетам.
│
├─ JournalViewer.exe - Программа для просмотра Контрольных Лент.
├─ FiscalConsole.exe - Сервисная программа ФМ
├─ FisUSBMRom.exe - Программа для считывания данных из ФМ
├─ FiscalUnit.dll - Драйвер Фискального Модуля.
├─ Ftd2xx.dll - Интерфейсная библиотека для работы с драйвером FiscalUnit.
├─ Hash.dll - Модуль функций вычисления хешей.
│
├─ Driver-FTDI\ - Раздел, в котором должен находиться FTDI драйвер.
├─ FisInfo.ini - Файл параметров инициализации.
├─ FisUnit.ini - Файл конфигурации ФМ FisUSBM
├─ Logs\ - Раздел, для хранения файлов протоколов.
│  ├── fisunit_DDMMYYYY_fisunit.log - Лог-файл обмена с FiscalUnit.dll
│  └── FisUSBM_DDMMYYYY.log - Лог-файл обмена с ФМ FisUSBM
│
└─ JournalViewer.cfg - Файл конфигурации JournalViewer.exe

```

3.2. ФАЙЛ ИНИЦИАЛИЗАЦИИ FISUNIT.INI

- Файл инициализации представляет собой обычный текстовый файл, содержащий набор параметров для работы Фискального Решения **FisUSBM**.

```

[Log]
Log=1 - Включение лог-файла обмена между FisUnit.dll и внешней программой Ext.Software (fisunit_DDMMYYYY_fisunit.log).
LogCount=30 - Количество одновременно хранимых лог файлов

[Port]
port=usb - Порт подключения ФМ FisUSBM к компьютеру (usb, com1, com2...)
;baud=115200 - Скорость обмена с модулем, если он подключен к COM-порту
timeoutRead=5000 - Максимальное время ожидания FisUSB при чтении (в миллисекундах)
timeoutSend=1000 - Максимальное время ожидания FisUSB при записи (в миллисекундах)

```

ВНИМАНИЕ!!! Не изменяйте без крайней необходимости параметры *timeoutRead* и *timeoutSend*.

```

[Journal]
WorkingDir=Journal - Имя раздела для текущего журнала, файла данных и лога аудита

```

PathCopy=Journal\Archive - Путь к разделу для хранения архива журналов и файлов данных
 PathCopy2=D:\Archive2 - Путь к разделу для хранения копии архива журналов и файлов данных.

ВНИМАНИЕ!!! Раздел для хранения копий архива Archive2 должен располагаться на отдельном носителе, вне корпуса POS-а. Например, на флешке, внешнем диске, другом компьютере и т.д.

[Printer]
 port=com2 - Последовательный порт, к которому подключен чековый принтер.
 baud=115200 - Скорость работы порта принтера
 timeout=1000 - Максимальное время ожидания ответа от принтера
 BufferDir=Journal\FMB - Путь к разделу для хранения буферов обмена принтера

[Display]
 port=com3 - Последовательный порт, к которому подключен дисплей покупателя
 baud=9600 - Скорость обмена порта дисплея
 timeout=1000 - Максимальное время ожидания ответа от дисплея (в миллисекундах)

3.2.1. ПРИМЕРЫ ФАЙЛОВ ПРОТОКОЛОВ И АУДИТА

Audit.log – Фрагмент файла аудита.

```

File  Правка  Вид  Кодировка  Справка  54 %
11-11-2016 16:41:08 Darbinieks Lena LOGIN
11-11-2016 16:41:19 Darbinieks Lena LOGOUT
11-11-2016 16:41:30 Darbinieks Lena LOGIN
11-11-2016 16:41:34 Darbinieks Lena LOGOUT
11-11-2016 16:41:34 Darbinieks Lena LOGIN
11-11-2016 16:41:51 Darbinieks Lena LOGOUT
11-11-2016 16:41:55 Darbinieks Lena LOGIN
11-11-2016 16:41:55 Darbinieks Lena LOGOUT
11-11-2016 16:42:57 Darbinieks Lena LOGIN
11-11-2016 16:43:13 Darbinieks Lena LOGOUT
11-11-2016 16:43:13 Uzņēnuma administrators Lena LOGIN
11-11-2016 16:43:13 Uzņēnuma administrators Lena LOGOUT
11-11-2016 16:43:32 Darbinieks Lena LOGIN
11-11-2016 16:43:44 Darbinieks Lena LOGOUT
11-11-2016 16:43:44 Virtualizācijas administrators Lena LOGIN
11-11-2016 16:43:55 Virtualizācijas administrators Lena LOGOUT
11-11-2016 16:44:09 Darbinieks Lena LOGIN
11-11-2016 16:44:09 Darbinieks Lena LOGOUT
11-11-2016 16:45:29 Darbinieks Lena LOGIN
11-11-2016 16:45:37 Darbinieks Lena LOGOUT
  
```

Fisunit_DDMMYYYY_fisunit.log – Фрагмент лог-файла обмена между FisUnit.dll и внешней программой.

```

File  Правка  Формат  Вид  Справка
28.10.2016 13:38:02 < 02 07 4F 1C 31 31 1C 48 03
28.10.2016 13:38:02 > 02 10 00 1C 41 1C 1C 41 1C 1C 41 1C 1C
0001: 0C 03
28.10.2016 13:38:06 @ GetFiscalInfoEx
28.10.2016 13:38:06 < 02 07 4F 1C 31 31 1C 48 03
28.10.2016 13:38:06 > 02 10 00 1C 41 1C 1C 41 1C 1C 41 1C 1C
0001: 0C 03
28.10.2016 13:38:08 @ GetFiscalInfoEx
28.10.2016 13:38:08 < 02 07 4F 1C 31 30 1C 49 03
28.10.2016 13:38:08 > 02 14 00 1C 41 1C 1C 41 1C 1C 41 1C 1C
0001: 1C 1C 1C 1C 08 03
28.10.2016 13:38:47 @ GetFiscalInfoEx
28.10.2016 13:38:53 < 02 07 4F 1C 31 30 1C 49 03
28.10.2016 13:39:02 > 02 14 00 1C 41 1C 1C 41 1C 1C 41 1C 1C
0001: 1C 1C 1C 1C 08 03
28.10.2016 13:40:34 @ GetFiscalInfoEx
28.10.2016 13:40:34 < 02 07 4F 1C 31 30 1C 49 03
28.10.2016 13:40:34 > 02 14 00 1C 41 1C 1C 41 1C 1C 41 1C 1C
0001: 1C 1C 1C 1C 08 03
28.10.2016 13:40:36 @ GetFiscalInfoEx
28.10.2016 13:40:36 < 02 07 4F 1C 31 31 1C 48 03
28.10.2016 13:40:36 > 02 10 00 1C 41 1C 1C 41 1C 1C 41 1C 1C
0001: 0C 03
  
```

3.3. ПРОГРАММА FisUSBROM.EXE

3.3.1. НАЗНАЧЕНИЕ ПРОГРАММЫ

- Программа предназначена для чтения, просмотра, и сохранения данных ФМ FisUSBM и позволяет:
 - Считать и сохранить в файл копию внутренней программы ФМ (Firmware).
 - Посчитать Hash-код внутренней программы.
 - Оценить объем оставшегося свободного места в памяти ФМ.
 - Прочитать из ФМ и сохранить в файле данные о Z-Отчетах и Чеках.

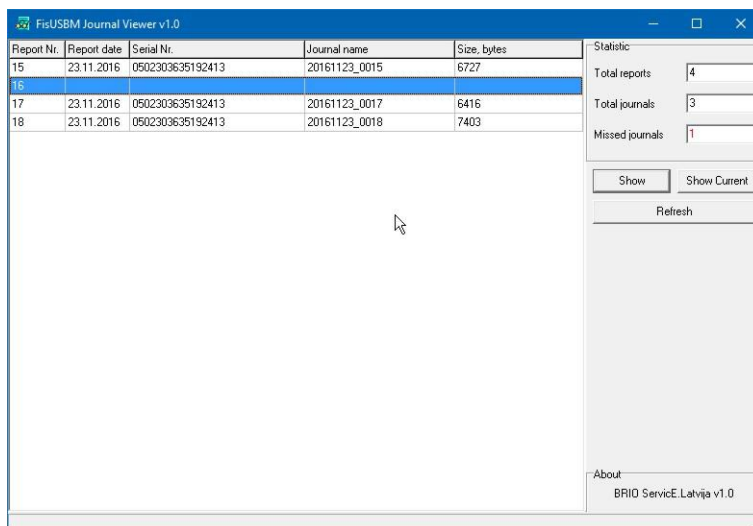
3.4. ПРОГРАММА JOURNAL VIEWER.EXE

3.4.1. НАЗНАЧЕНИЕ И НАСТРОЙКИ

- Программа предназначена для работы с архивами электронных контрольных лент ФР и позволяет:
 - Просматривать контрольные ленты.
 - Проверять их целостность.
 - Осуществлять поиск нужного текста в лентах.
 - Рассчитывать HASH SHA-1 для выделенного фрагмента текста контрольной ленты.
- Программа работает непосредственно с файлами архивов контрольных лент, поэтому наличие в системе ФМ FisUSBM не обязательно.

ВНИМАНИЕ!!! Местоположение просматриваемого файла архива определяется параметром `PathCopy=Journal\Archive` в файле инициализации `FisUnit.ini`.

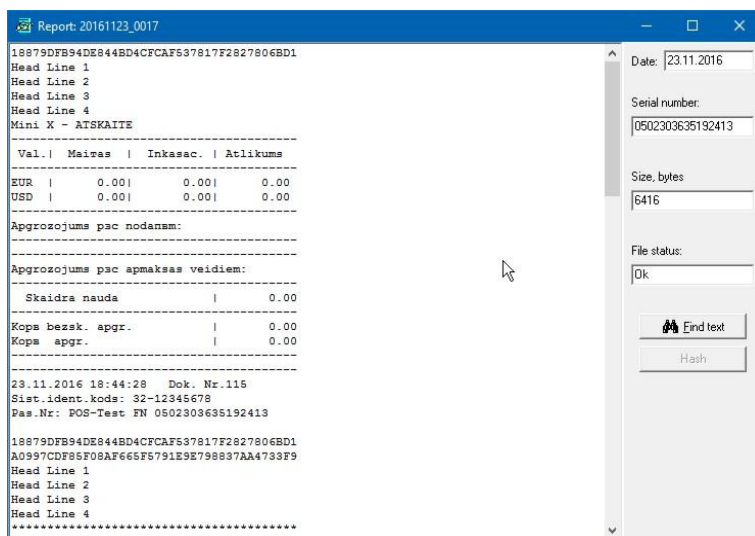
3.4.2. ГЛАВНОЕ ОКНО ПРОГРАММЫ



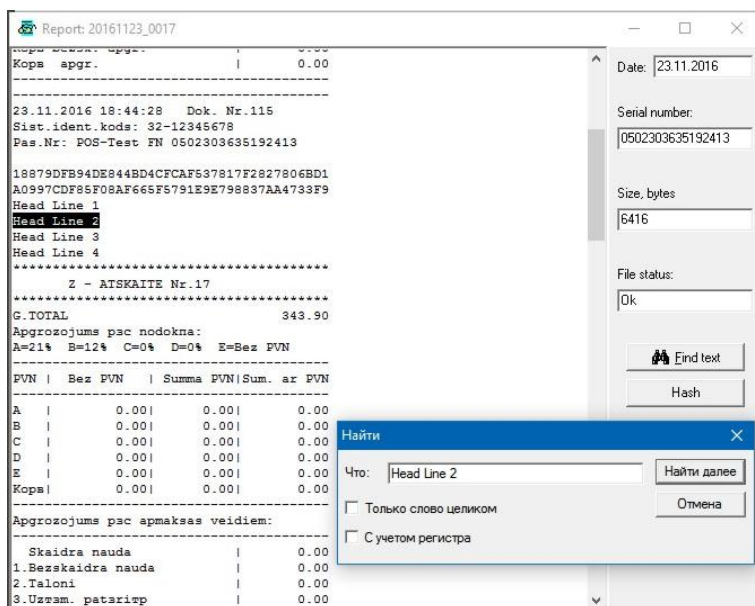
- На рисунке – главная экранная форма программы JournalViewer.
- В левой части экранной формы – окно со списком контрольных лент, хранящиеся на жестком диске компьютера в разделе /Journal/Archive. Если какая либо лента отсутствует в архиве, то ее позиция будет обозначена пустой строкой.
 - **Report Nr.:** – Порядковый номер контрольной ленты.
 - **Report date:** – Дата создания ленты.
 - **Serial Nr.:** – Серийный номер ФМ, создавшего ленту.
 - **Journal name:** – Название ленты в формате: ДАТА (YYYYMMDD)_ НОМЕР ЛЕНТЫ

- **Size bytes:** – Размер файла с лентой в байтах.
- В правой части формы – поля с текущей информацией о контрольных лентах.
 - **Statistic/Total reports:** – Общее количество контрольных лент, которые должны присутствовать в разделе /Archive.
 - **Statistic/Total journals:** – Фактическое количество найденных контрольных лент.
 - **Statistic/Missed journals:** – Общее количество отсутствующих лент.
 - **[Refresh]:** – Обновить информацию о контрольных лентах из раздела /Archive.
 - **[Show]:** – Показать содержимое выбранной контрольной ленты. Команда дублируется двойным щелчком курсора в соответствующей строке главного окна.
 - **[Show Current]:** - Показать содержимое текущей (Еще не архивированной) контрольной ленты.

3.4.3. ПРОСМОТР И ПРОВЕРКА ЛЕНТЫ

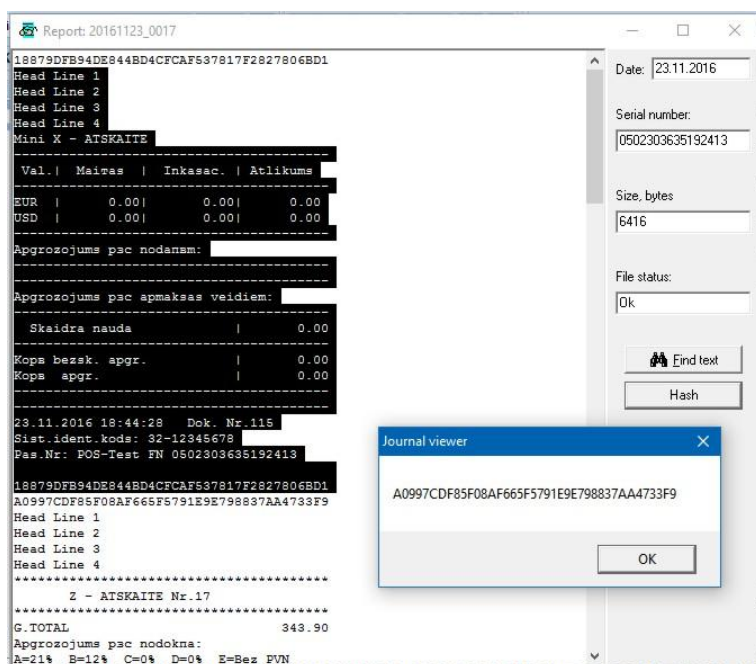


- На рисунке - экранная форма с контрольной лентой, выбранной по команде **Statistic / [Show]**.
- В левой части формы – окно с содержимым контрольной ленты.
- В правой части формы - поля с данными о ленте и клавиши команд.
 - **Date** – Дата создания ленты.
 - **Serial Nr.** – Серийный номер ФМ, создавшего эту ленту.
 - **Size bytes** – Размер файла с лентой в байтах.
 - **File status** – Результат проверки целостности файла.
- Что бы найти нужный текст в ленте, следует нажать клавишу **[Find Text]** и в открывшейся вспомогательной форме ввести искомый фрагмент текста.



3.4.4. РАСЧЕТ ЗНАЧЕНИЯ HASH SHA-1 ДЛЯ ФРАГМЕНТА ЛЕНТЫ

- Что бы посчитать значение **HASH SHA-1** для фрагмента текста контрольной ленты, необходимо выделить нужный фрагмент курсором и нажать на клавишу **[Hash]**.



- На рисунке показан результат расчета значения **HASH SHA-1** для выделенного фрагмента текста Контрольной Ленты, содержащего один Чек (От первой строки заголовка, до последней строки, содержащей значение **HASH SHA-1** предыдущего Чека).
- Как видно в примере - посчитанное значение **HASH SHA-1** фрагмента текста с Чеком полностью совпадает со значением **HASH SHA-1** этого же Чека, записанное в первой строке следующего Чека контрольной ленты. Следовательно, проверенный Чек правильный и он не подвергался каким либо изменениям.

3.4.5. ПРИМЕРЫ ФРАГМЕНТОВ ФАЙЛОВ

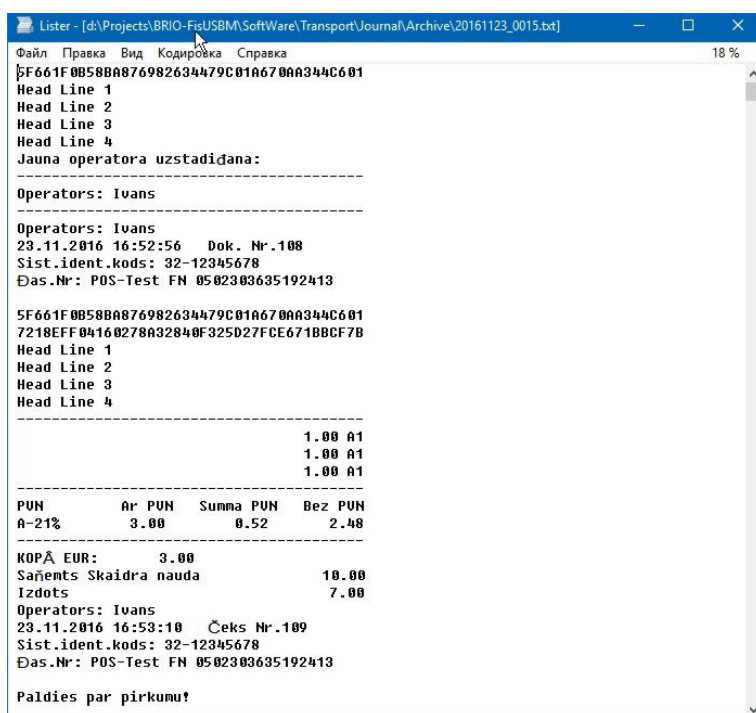
YYYYMMDD_SerialNumber.hdr – Архивный контрольный файл Электронной Ленты.



```

Listner - [d:\Projects\BRIO-FisUSBM\SoftWare\Transport\Journal\Archive\20161123_0015.hdr]
Файл  Правка  Вид  Кодировка  Справка  100 %
[Head]
CRC=5E98E413
Time=1653
Date=20161123
SerialNumber=0502303635192413
CS=AFFF1E58
  
```

YYYYMMDD_SerialNumber.txt – Архивный файл Электронной Контрольной Ленты.



```

Listner - [d:\Projects\BRIO-FisUSBM\SoftWare\Transport\Journal\Archive\20161123_0015.txt]
Файл  Правка  Вид  Кодировка  Справка  18 %
5F661F0B58BA876982634479C01A670AA344C601
Head Line 1
Head Line 2
Head Line 3
Head Line 4
Jauna operatora uzstaidana:
-----
Operators: Ivans
-----
Operators: Ivans
23.11.2016 16:52:56 Dok. Nr.108
Sist.ident.kods: 32-12345678
Das.Nr: POS-Test FN 0502303635192413

5F661F0B58BA876982634479C01A670AA344C601
7218EFF04160278A32840F325D27FCE671B8CF7B
Head Line 1
Head Line 2
Head Line 3
Head Line 4

-----
1.00 A1
1.00 A1
1.00 A1
-----
PUN      Ar PUN  Summa PUN  Bez PUN
A-21%    3.00   0.52      2.48
-----
KOPĀ EUR:    3.00
Saņemts Skaidra nauda          10.00
Izdots                          7.00
Operators: Ivans
23.11.2016 16:53:10 Čeks Nr.109
Sist.ident.kods: 32-12345678
Das.Nr: POS-Test FN 0502303635192413

Paldies par pirkumu!
  
```

4. ПРИЛОЖЕНИЯ

4.1. БИБЛИОТЕКА ФУНКЦИЙ FiscalUNIT.

4.1.1. УСТАНОВОЧНЫЕ ФУНКЦИИ

SetHeaderEx2

Прототип:

```
int CALLBACK SetHeaderEx2( int attr1, char *line1, int attr2, char *line2, int attr3, char *line3, int attr4, char *line4, int attr5, char *line5, int attr6, char *line6 )
```

Описание:

Функция позволяет задать шесть строк, которые будут печататься в начале каждого чека. Размер строк не может быть больше 40 символов.

Параметры:

| | |
|-------|--|
| attr1 | атрибуты печати первой строки |
| line1 | первая строка |
| attr2 | атрибуты печати второй строки |
| line2 | вторая строка |
| attr3 | атрибуты печати третьей строки |
| line3 | третья строка |
| attr4 | атрибуты печати четвертой строки |
| line4 | четвертая строка |
| attr5 | атрибуты печати пятой строки |
| line5 | пятая строка |
| attr6 | атрибуты печати шестой строки |
| line6 | шестая строка |

SetFooter

Прототип:

```
int CALLBACK SetFooter( int attr1, char *line1, int attr2, char *line2, int attr3, char *line3, int attr4, char *line4 )
```

Описание:

Функция позволяет задать четыре строки, которые будут печататься в конце каждого чека. Размер строк не может быть больше 40 символов.

Параметры:

| | |
|-------|--|
| attr1 | атрибуты печати первой строки |
| line1 | первая строка |
| attr2 | атрибуты печати второй строки |
| line2 | вторая строка |
| attr3 | атрибуты печати третьей строки |
| line3 | третья строка |
| attr4 | атрибуты печати четвертой строки |
| line4 | четвертая строка |

EnableFooter

Прототип:

```
int CALLBACK EnableFooter( int enable )
```

Описание:

Функция разрешает, или запрещает печатать в конце чека текст, который введен с помощью функции [SetFooter](#).

Параметры:

| | |
|--------|--|
| enable | 0 – печать запрещена, 1 – печать разрешена |
|--------|--|

SetFooter2

Прототип:

```
int CALLBACK SetFooter2( int attr1, char *line1, int attr2, char *line2 )
```

Описание:

Функция позволяет задать две строки, которые будут печататься в конце каждого чека. Размер строк не может быть больше 40 символов.

Параметры:

| | |
|-------|---|
| attr1 | атрибуты печати первой строки |
| line1 | первая строка |
| attr2 | атрибуты печати второй строки |
| line2 | вторая строка |

EnableFooter2**Прототип:**

```
int CALLBACK EnableFooter2( int enable )
```

Описание:

Функция разрешает или запрещает печатать в конце чека текст, который введен с помощью функции [SetFooter2](#).

Параметры:

| | |
|--------|--|
| enable | 0 – печать запрещена, 1 – печать разрешена |
|--------|--|

SetPrinterTypeEx2**Прототип:**

```
int CALLBACK SetPrinterTypeEx2( int printerType, int coding, int image, int imagenumber, int imagelarge )
```

Описание:

Установка типа и параметров принтера.

Параметры:

| | |
|-------------|--|
| printerType | тип принтера |
| coding | таблица кодировки (0 – DOS, 1 – WIN-1257) |
| image | 0 – печать логотипа выключена, 1 – включена |
| imagenumber | номер изображения логотипа (допуск 1..8) |
| imagelarge | 0 – печать логотипа в узком формате, 1 – в широком формате |

SetCompressionMode**Прототип:**

```
int CALLBACK SetCompressionMode( int compression )
```

Описание:

Функция позволяет устанавливать интервал между строками текста на принтере.

Параметры:

| | |
|-------------|---|
| compression | интервал между строками (допуск 0..100, рекомендуется 15) |
|-------------|---|

SetTillImpuls**Прототип:**

```
int CALLBACK SetTillImpuls( int time1, int time2 )
```

Описание:

Функция позволяет устанавливать длительность импульса для открытия денежного ящика.

Параметры:

| | |
|-------|---|
| time1 | десятки миллисекунд ON состояния (допуск 0..100, рекомендуется 40) |
| time2 | десятки миллисекунд OFF состояния (допуск 0..100, рекомендуется 40) |

SetDepartName

Прототип:
int CALLBACK SetDepartName(int number, char *name)

Описание:
Установка названий отделов.

Параметры:

| | |
|--------|--|
| number | номер отдела (допуск 0..7) |
| name | название отдела (максимум 15 символов) |

SetCredit

Прототип:
int CALLBACK SetCredit(int number, char *name)

Описание:
Установка названий видов безналичной оплаты.

Параметры:

| | |
|--------|---|
| number | номер вида оплаты (допуск 0..7) |
| name | название вида оплаты (максимум 15 символов) |

SetCurrency

Прототип:
int CALLBACK SetCurrency(int number, char *name, double rate)

Описание:
Установка курсов валют. Если значение курса равно 0, то данная валюта будет заблокирована.

Параметры:

| | |
|--------|---|
| number | номер валюты (допуск 0..3) |
| name | название валюты (максимум 4 символа) |
| rate | курс валюты (для валюты 0 курс всегда должен равняться 1) |

SetConvertType

Прототип:
int CALLBACK SetConvertType(bool type)

Описание:
Установка типа конвертации валюты.

Параметры:

| | |
|------|--|
| type | false – прямая конвертация, true – косвенная |
|------|--|

SetVAT

Прототип:
int CALLBACK SetVAT(int number, double rate)

Описание:
Установка ставок налогов. Всего можно установить четыре ставки налога. Пятая ставка недоступна для установки, всегда равно 0 и предназначена для продажи товаров БАЗ НАЛОГА.

Параметры:

| | |
|--------|----------------------------|
| number | номер налога (допуск 0..3) |
| rate | ставка налога |

AllowGoodsReturn

Прототип:
int CALLBACK AllowGoodsReturn(char *unitNumber)

Описание:

Разрешение возврата товаров. Если в фискальный модуль отправляется правильный серийный номер, то возврат товаров - разрешен. Если требуется запретить возврат товаром, то следует послать любой набор цифр, отличный от правильного серийного номера.

Параметры:

unitNumber серийный номер данного экземпляра фискального модуля

4.1.2. ИНФОРМАЦИОННЫЕ ФУНКЦИИ

GetFiscalInfo

Прототип:

int CALLBACK GetFiscalInfo(int infoType, char *data)

Описание:

Получение информации. Возвращаемые поля данных, если их больше одного, разделяются символом 0x1C.

Параметры:

infoType [код](#) запрашиваемой информации

data буфер для запрашиваемой информации

PrintVATTable

Прототип:

int CALLBACK PrintVATTable(void)

Описание:

Распечатать таблицу налогов.

PrintCurrencyTable

Прототип:

int CALLBACK PrintCurrencyTable(void)

Описание:

Распечатать курсы валют.

PrintCreditTable

Прототип:

int CALLBACK PrintCreditTable(void)

Описание:

Распечатать таблицу видов безналичной оплаты.

PrintSumPeriodicReport

Прототип:

int CALLBACK PrintSumPeriodicReport(char *date1, char *date2)

Описание:

Распечатать суммарный фискальный отчет за период.

Параметры:

date1 начальная дата периода (формат YYYYMMDD)

date2 конечная дата периода (формат YYYYMMDD)

PrintPeriodicReport

Прототип:

int CALLBACK PrintPeriodicReport(char *date1, char *date2)

Описание:

Распечатать все Z-отчеты за период дат.

Параметры:

date1 начальная дата периода (формат YYYYMMDD)
date2 конечная дата периода (формат YYYYMMDD)

PrintSumPeriodicReportByNumber*Прототип:*

int CALLBACK PrintSumPeriodicReportByNumber(int number1, int number2)

Описание:

Распечатать суммарный фискальный отчет в диапазоне номеров.

Параметры:

number1 начальный номер Z-отчёта (формат YYYYMMDD)
number2 конечный номер Z-отчёта (формат YYYYMMDD)

PrintPeriodicReportByNumber*Прототип:*

int CALLBACK PrintPeriodicReportByNumber(int number1, int number2)

Описание:

Распечатать все Z-отчеты в диапазоне номеров.

Параметры:

number1 начальный номер отчёта (формат YYYYMMDD)
number2 конечный номер отчёта (формат YYYYMMDD)

4.1.3. СИСТЕМНЫЕ ФУНКЦИИ

SetClerk **(Добавлены параметры!)***Прототип:*

int CALLBACK SetClerk(int attr1, char *line1, **int type, bool print, bool login**)

Описание:

Регистрация пользователя.

Параметры:

attr1 [атрибуты](#) печати имени пользователя
line1 имя пользователя
type [тип](#) пользователя
print разрешение печати чека регистрации
login тип регистрации пользователя: true – вход, false – выход

ResetFiscal*Прототип:*

int CALLBACK ResetFiscal(void)

Описание:

Сброс ФМ. Функция переключает модуль в состояние IDLE. Если есть незавершенный чек - он аннулируется. Эта функция не выполняется в состоянии HARD. Обычно эта функция используется, что бы отменить начатую печать фискального, или не фискального чека.

MoneyInCurr*Прототип:*

int CALLBACK MoneyInCurr(int number, double money)

Описание:

Внесение наличных денег.

Параметры:

number номер валюты (допуск 0..3)
money сумма вносимых денег в указанной валюте

MoneyOutCurr*Прототип:*

int CALLBACK MoneyOutCurr(int number, double money)

Описание:

Инкассация.

Параметры:

number номер валюты (допуск 0..3)
money сумма инкассируемых денег в указанной валюте

CustomerDisplay*Прототип:*

int CALLBACK CustomerDisplay(int displayType, char *line1, char *line2)

Описание:

Вывод информации на дисплей покупателя.

Параметры:

displayType зарезервированный параметр (допускается любое значение)
line1 строка 1 (максимум 20 символов)
line2 строка 2 (максимум 20 символов)

SetDispParam*Прототип:*

int CALLBACK SetDispParam(int code, int type)

Описание:

Установка параметров дисплея покупателя.

Параметры:

code таблица кодировки (0 – DOS, 1 – WIN-1257)
type тип (0 – DEC, 1 – HEX)

SetDispTime*Прототип:*

int CALLBACK SetDispTime(void)

Описание:

Переключение дисплея покупателя в режим отображения текущего времени.

OpenCashDrawer*Прототип:*

int CALLBACK OpenCashDrawer(void)

Описание:

Открыть денежный ящик.

4.1.4. ФУНКЦИИ ФИСКАЛЬНЫХ ЧЕКОВ

BeginFiscalReceipt*Прототип:*

int CALLBACK BeginFiscalReceipt(void)

Описание:

Начать фискальный чек.

PrintRecltemDepart **(Добавлены параметры!)**

Прототип:

```
int CALLBACK PrintRecltemDepart( char *name, double quantity, double price, int taxNumber, char
*unit,
int depart, char *code )
```

Описание:

Добавить товарную позицию в чек.

Параметры:

| | |
|-----------|--|
| name | название товара (максимум 29 символов) |
| quantity | количество товара |
| price | цена товара |
| taxNumber | номер налога (допуск 0..4) |
| unit | название единицы измерения количества товара |
| depart | номер отдела (допуск 0..7) |
| code | код товара |

DiscountAdditionForItem

Прототип:

```
int CALLBACK DiscountAdditionForItem( int type, double val )
```

Описание:

Задать скидку или наценку на последнюю введённую товарную позицию в чеке.

Параметры:

| | |
|------|--|
| type | тип скидки/наценки (1 – скидка в процентах, 2 – скидка в деньгах) |
| val | значение скидки (отрицательное значение – скидка, положительное – наценка) |

ItemReturnEx **(Добавлены параметры!)**

Прототип:

```
int CALLBACK ItemReturnEx( char *name, double quantity, double price, int taxNumber, char *unit,
int depart, int discountType, double discount, char *code )
```

Описание:

Отмена товарной позиции в чеке.

Параметры:

| | |
|--------------|--|
| name | название товара (максимум 29 символов) |
| quantity | количество товара |
| price | цена товара |
| taxNumber | номер налога (допуск 0..4) |
| unit | название единицы измерения количества товара |
| depart | номер отдела (допуск 0..7) |
| discountType | тип скидки/наценки (1 – скидка в процентах, 2 – скидка в деньгах) |
| discount | значение скидки (отрицательное значение – скидка, положительное – наценка) |
| code | код товара |

DiscountAdditionForReceipt

Прототип:

```
int CALLBACK DiscountAdditionForReceipt( int type, double val )
```

Описание:

Скидка на весь чек.

Параметры:

| | |
|------|--|
| type | тип скидки/наценки (1 – скидка в процентах, 2 – скидка в деньгах) |
| val | значение скидки (отрицательное значение – скидка, положительное – наценка) |

PrintCommentLine*Прототип:*

int CALLBACK PrintCommentLine(char *line, int attr)

Описание:

Распечатать строку текста в фискальном чеке. Например, комментарий или разделительные линии.

Параметры:

line текст (максимум 30 символов)
attr [атрибуты](#) печати текста

ClientInfo (Новая функция!)*Прототип:*

int CALLBACK ClientInfo(char *regNr, char *vatNr, char *name, char *addr)

Описание:

Добавить в чек реквизиты покупателя.

Параметры:

regNr регистрационный номер (максимум 11 символов)
vatNr НДС номер (максимум 13 символов)
name название
addr адрес

EndFiscalReceiptCurrEx*Прототип:*

```
int CALLBACK EndFiscalReceiptCurrEx( double summCash, double summCredit1, double summCredit2,  
double summCredit3, double summCredit4, double summCredit5, double summCredit6, double summCredit7,  
double summCredit8, double summCur1, double summCur2, double summCur3 )
```

Описание:

Завершить чек продажи.

Параметры:

summCash полученная сумма наличными в базовой валюте
summCredit1 полученная сумма безналичным видом оплаты 1
summCredit2 полученная сумма безналичным видом оплаты 2
summCredit3 полученная сумма безналичным видом оплаты 3
summCredit4 полученная сумма безналичным видом оплаты 4
summCredit5 полученная сумма безналичным видом оплаты 5
summCredit6 полученная сумма безналичным видом оплаты 6
summCredit7 полученная сумма безналичным видом оплаты 7
summCredit8 полученная сумма безналичным видом оплаты 8
summCur1 полученная сумма наличными в валюте 1
summCur2 полученная сумма наличными в валюте 2
summCur3 полученная сумма наличными в валюте 3

GoodsReturnCurrEx*Прототип:*

```
int CALLBACK GoodsReturnCurrEx( double summCash, double summCredit1, double summCredit2,
double summCredit3, double summCredit4, double summCredit5, double summCredit6, double
summCredit7,
double summCredit8, double summCur1, double summCur2, double summCur3 )
```

Описание:

Завершить чек возврата.

Параметры:

| | |
|-------------|---|
| summCash | возвращённая сумма наличными в базовой валюте |
| summCredit1 | возвращённая сумма безналичным видом оплаты 1 |
| summCredit2 | возвращённая сумма безналичным видом оплаты 2 |
| summCredit3 | возвращённая сумма безналичным видом оплаты 3 |
| summCredit4 | возвращённая сумма безналичным видом оплаты 4 |
| summCredit5 | возвращённая сумма безналичным видом оплаты 5 |
| summCredit6 | возвращённая сумма безналичным видом оплаты 6 |
| summCredit7 | возвращённая сумма безналичным видом оплаты 7 |
| summCredit8 | возвращённая сумма безналичным видом оплаты 8 |
| summCur1 | возвращённая сумма наличными в валюте 1 |
| summCur2 | возвращённая сумма наличными в валюте 2 |
| summCur3 | возвращённая сумма наличными в валюте 3 |

4.1.5. ФУНКЦИИ НЕ ФИСКАЛЬНЫХ ЧЕКОВ

BeginNonFiscalReceipt

Прототип:

```
int CALLBACK BeginNonFiscalReceipt( void )
```

Описание:

Начать не фискальный чек.

PrintNonFiscalLine

Прототип:

```
int CALLBACK PrintNonFiscalLine( char *line, int attr )
```

Описание:

Распечатать строку текста в не фискальном чеке. Например, комментарий или разделительные линии.

Параметры:

| | |
|------|--|
| line | текст (максимум 40 символов) |
| attr | атрибуты печати текста |

PrintTareItem

Прототип:

```
int CALLBACK PrintTareItem( char *name, double quantity, double price )
```

Описание:

Добавить в чек позицию приёма тары.

Параметры:

| | |
|----------|--------------------------------------|
| name | название тары (максимум 29 символов) |
| quantity | количество принимаемой тары |
| price | цена принимаемой тары за единицу |

PrintTareItemVoid

Прототип:

```
int CALLBACK PrintTareItemVoid( char *name, double quantity, double price )
```

Описание:

Добавить в чек позицию выдачи тары.

Параметры:

| | |
|----------|--------------------------------------|
| name | название тары (максимум 29 символов) |
| quantity | количество выдаваемой тары |
| price | цена выдаваемой тары за единицу |

PrintDepositReceive*Прототип:*

```
int CALLBACK PrintDepositReceive( char *name, double quantity, double price )
```

Описание:

Добавить в чек позицию приёма депозита.

Параметры:

| | |
|----------|--------------------------|
| name | название депозита |
| quantity | количество депозита |
| price | цена депозита за единицу |

PrintDepositRefund*Прототип:*

```
int CALLBACK PrintDepositRefund( char *name, double quantity, double price )
```

Описание:

Добавить в чек позицию возврата депозита.

Параметры:

| | |
|----------|--------------------------|
| name | название депозита |
| quantity | количество депозита |
| price | цена депозита за единицу |

EndNonFiscalReceipt*Прототип:*

```
int CALLBACK EndNonFiscalReceipt( void )
```

Описание:

Завершить печать не фискального чека.

PrintHeader*Прототип:*

```
int CALLBACK PrintHeader( void )
```

Описание:

Распечатать шапку.

PrintBarCodeTG2480*Прототип:*

```
int CALLBACK PrintBarCodeTG2480( int barType, int barHRI, int barHeight, int barWidth, char *barCode )
```

Описание:

Распечатать штрих-код на принтере TG-2480.

Параметры:

| | |
|-----------|--|
| barType | тип штрих-кода |
| barHRI | печать HRI (Human Readable Interpretation) (Bit ₀ – сверху, Bit ₁ – снизу) |
| barHeight | высота штрих-кода (допуск 1..255) |
| barWidth | ширина штрих-кода (допуск 1..6) |
| barCode | штрих-код |

PrintQRCode*Прототип:*

int CALLBACK PrintQRCode(int qrSize, int qrAlignment, char * qrData)

Описание:

Распечатать QR-код.

Параметры:

| | |
|-------------|---|
| qrSize | размер QR-кода (допуск 1..8) |
| qrAlignment | расположение (0 – слева, 1 – по центру, 2 – справа) |
| qrData | данные QR-кода (максимум 40 символов) |

4.1.6. СЕРВИСНЫЕ ФУНКЦИИ

Fiscalization*Прототип:*

int CALLBACK Fiscalization(char *date)

Описание:

Активация фискального режима.

Параметры:

date дата активации (формат YYYYMMDD)

SetId*Прототип:*

int CALLBACK SetId(char *id)

Описание:

Установка идентификационного номера.

Параметры:

id идентификационный номер (максимум 16 символов)

SetVoidName*Прототип:*

int CALLBACK SetVoidName(int type)

Описание:

Установка названия коррекции (0 – коррекция, 1 – скидка, 2 – пустота), или наценки (10 – наценка, 11 – услуга).

Параметры:

type значения параметра указано в описании

4.1.7. АТРИБУТЫ ПЕЧАТИ

| | Значение - 0 | Значение - 1 |
|------------------|----------------------|------------------------|
| Bit ₀ | Размер шрифта 9x9 | Размер шрифта 7x9 |
| Bit ₁ | Всегда | Запрещённое значение |
| Bit ₂ | - | Подчёркнутый |
| Bit ₃ | - | Полужирный |
| Bit ₄ | - | Двойная высота |
| Bit ₅ | - | Двойная ширина |
| it ₆ | Запрещенное значение | Всегда |
| it ₇ | - | Выравнивание по центру |

4.1.8. КОДЫ ЗАПРАШИВАЕМОЙ ИНФОРМАЦИИ

| Номер | Константа | Описание: |
|----------|-------------------------|--|
| 0 | infoReceiptSumm | Итоговая сумма текущего фискального чека |
| 1 | infoShiftTurnover | Дневной оборот |
| 2 | infoReceiptNumber | Номер следующего чека |
| 3 | infoUnitNumber | Номер фискального модуля |
| 4 | infoUnitVersion | Версия ПО фискального модуля |
| 7 | infoReportNumber | Номер текущего Z-отчёта |
| 8 | infoDayReceiptNumber | Номер следующего чека за данный день |
| 9 | infoUnitState | Код состояния фискального модуля |
| 10 | infoHeader | Шапка |
| 11 | infoFooter | Нижние строки чека |
| 12 | infoFooter2 | Нижние строки чека |
| 13 | infoFooterEnabled | Разрешение печати нижних строк чека |
| 14 | infoFooter2Enabled | Разрешение печати нижних строк чека |
| 30..33 | infoCurDescription+N | Название и курс валюты N |
| 40..43 | infoTax+N | Ставка налога N |
| 50..53 | infoCurCash+N | Сумма наличных денег в ящике валюты N |
| 60..67 | infoCreditDescription+N | Название безналичного вида оплаты N |
| 301 | infoShassiNumber | Номер шасси |
| 302 | infoId | Идентификационный номер |
| 306 | infoConvertType | Тип конвертации валюты |
| 307 | infoTG2480Status | Статус принтера TG-2480 |
| 310..317 | infoDepartDescription+N | Информация по отделу N |

4.1.9. ТИПЫ ПРИНТЕРОВ

| Номер | Тип |
|-------|---------------------------|
| 0 | Epson 210 |
| 1 | STAR TYPE 900 |
| 2 | Epson 260 |
| 3 | BA-T500II |
| 4 | TG2480 |
| 5 | Reserved 5 |
| 6 | Reserved 6 |
| 7 | Reserved 7 |
| 8 | Reserved 8 |
| 9 | Samsung SRP 350(57mm) |
| 10 | Samsung SRP 350 (thermal) |
| 11 | Samsung SRP 275 |

4.1.10. ТИПЫ ПОЛЬЗОВАТЕЛЕЙ

| Номер | Тип |
|-------|----------------------------------|
| 0 | Продавец |
| 1 | Администратор предприятия |
| 2 | Администратор виртуальной машины |
| 3 | Администратор |

4.1.11. КОДЫ СОСТОЯНИЯ МОДУЛЯ

| Номер | Состояние |
|-------|-----------|
| 0 | IDLE |
| 1 | NONFIS |
| 2 | FIS |
| 3 | HARD |
| 4 | ZIDLE |
| 5 | JCLOSE |

4.1.12. ТИПЫ ШТРИХ-КОДОВ

| Номер | Тип |
|-------|----------|
| 0 | UPC-A |
| 1 | UPC-E |
| 2 | EAN-13 |
| 3 | EAN-8 |
| 4 | CODE 39 |
| 5 | ITF |
| 6 | CODABAR |
| 7 | CODE 93 |
| 8 | CODE 128 |

4.2. КОДЫ ОШИБОК

4.2.1. ОШИБКИ ОТ FiscalUnit.DLL

| ОШИБКА | НОМЕР | | ПРИМЕЧАНИЕ |
|-----------------------|-------|------|---|
| | Dec | HEX | |
| ERR_Send | 160 | 0xA0 | Ошибка отправки пакета в ФМ |
| ERR_Read | 161 | 0xA1 | Ошибка получения пакета от ФМ (Таймаут) |
| ERR_Buffer_Too_Small | 163 | 0xA3 | Внутренняя ошибка (Неверный размер предоставленного буфера) |
| ERR_Wrong_Size | 164 | 0xA4 | Обмен с ФМ-Неверный размер пакета |
| ERR_Crc | 165 | 0xA5 | Обмен с ФМ-Неверная контрольная сумма в полученном пакете |
| ERR_Wrong_Format | 166 | 0xA6 | Обмен с ФМ - Неверный формат пакета |
| ERR_Open_Fiscal | 167 | 0xA7 | Ошибка подключения к ФМ (Чаще будет возникать Error_Send) |
| ERR_Bad_Parameter | 168 | 0xA8 | Обмен с ФМ - Полученное значение не совпадает с запрошенным значением |
| ERR_Make_Directory | 169 | 0xA9 | Ошибка создания раздела на диске |
| ERR_Send_Printer | 170 | 0xAA | Ошибка отправки данных на принтер (Ошибка печати) |
| ERR_Delete_File | 171 | 0xAB | Ошибка удаления файла |
| ERR_Open_File | 172 | 0xAC | Ошибка открытия файла |
| ERR_Read_File | 173 | 0xAD | Ошибка чтения файла |
| ERR_Write_File | 174 | 0xAE | Ошибка записи в файл |
| ERR_Create_File | 175 | 0xAF | Ошибка создания файла |
| ERR_Journal_Is_Lost | 176 | 0xB0 | Файл journal.txt не найден |
| ERR_Datafile_Is_Lost | 177 | 0xB1 | Файл database.csv не найден |
| ERR_Journal_Corrupted | 178 | 0xB2 | Нарушена структура файла journal.txt |
| ERR_Copy_File | 179 | 0xB3 | Ошибка копирования файла |

4.2.1. ОШИБКИ, ТРАНСЛИРУЕМЫЕ ОТ FisUSBM

| ОШИБКА | НОМЕР | | ПРИМЕЧАНИЕ |
|---------------------|-------|------|---|
| | DEC | HEX | |
| ERR_Not_Jmp | 201 | 0xC9 | Нет сервисной перемычки на плате |
| ERR_Write_Eeprom | 202 | 0xD2 | Ошибка записи в память |
| ERR_Flash_Write | 203 | 0xCB | Ошибка записи в флеш-память |
| ERR_Fuel_NrBlock | 204 | 0xCC | Недопустимый номер блока данных |
| ERR_Lenght | 205 | 0xCD | Неверная длина пакета |
| ERR_Stat_Fis | 206 | 0xCE | Недопустимая команда в фискальном режиме |
| ERR_Stat_Init | 207 | 0xCF | Недопустимая команда в инициализированном режиме |
| ERR_Date | 208 | 0xD0 | Неверный формат даты |
| ERR_Time | 209 | 0xD1 | Неверный формат времени |
| ERR_Stat_NoFis | 210 | 0xD2 | Недопустимая команда в нефискальном режиме |
| ERR_Num_Check | 211 | 0xD3 | Недопустимый номер чека. |
| ERR_GT_Check | 212 | 0xD4 | GT текущего чека меньше GT предыдущего чека. |
| ERR_Time_Check | 213 | 0xD5 | Время/Дата текущего чека меньше Время/Дата предыдущего чека. |
| ERR_Num_Z | 214 | 0xD6 | Недопустимый номер Z-отчета |
| ERR_GT_Z | 215 | 0xD7 | GT текущего Z-отчета меньше GT предыдущего Z-отчета |
| ERR_Time_Z | 216 | 0xD8 | Время/Дата текущего Z-отчета меньше Время/Дата предыдущего Z-отчета |
| ERR_Serial | 217 | 0xD9 | Серийный номер ФМ в команде |
| ERR_Too_Many_Bad | 218 | 0xDA | Слишком много плохих блоков в флеш-памяти |
| ERR_Not_Ident_Bad | 219 | 0xDB | |
| ERR_Unknown_Command | 220 | 0xDC | Неизвестная команда. |