



О. Халатов

Фискальный модуль Fis-FM32/32E(M)<sup>™</sup> для POS систем  
BRIO EngineerinG  
Рига, Латвия.



Данный документ содержит описание и порядок работы и программирования фискальных модулей BRIO-FisFM32<sup>™</sup>, BRIO-FisFM32E<sup>™</sup>, BRIO-FisFM32EM<sup>™</sup>.

**ВНИМАНИЕ!!! Сервисное обслуживание и настройки фискального модуля могут производиться только авторизованными сервисными центрами BRIO SRC!!!**

## ОГЛАВЛЕНИЕ

<b>1. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ.....</b>	<b>3</b>
<b>2. РАЗЪЕМЫ И ИНТЕРФЕЙСЫ .....</b>	<b>5</b>
2.1. РАСПОЛОЖЕНИЕ РАЗЪЕМОВ BRIO-FisFM32, BRIO-FisFM32E .....	5
2.1. РАСПОЛОЖЕНИЕ РАЗЪЕМОВ BRIO-FisFM32E(M) .....	5
2.2. ИНТЕРФЕЙС СВЯЗИ С КОМПЬЮТЕРОМ: RS-232C/USB .....	6
2.3. ИНТЕРФЕЙС ЧЕКОВОГО ПРИНТЕРА: RS-232C .....	6
2.1. ИНТЕРФЕЙС МОДУЛЯ VOX-FM32: VOX .....	6
2.1. ИНТЕРФЕЙС ДИСПЛЕЯ ПОКУПАТЕЛЯ: RS-232C .....	6
2.1. РАЗЪЕМ ДЛЯ ПОДКЛЮЧЕНИЯ КАБЕЛЯ ПИТАНИЯ ОТ РС .....	6
<b>3. ПОДКЛЮЧЕНИЕ ФИСКАЛЬНОГО МОДУЛЯ.....</b>	<b>7</b>
3.1. МОНТАЖ И ЗАЗЕМЛЕНИЕ КОРПУСА .....	7
3.2. ПИТАНИЕ ФИСКАЛЬНЫХ МОДУЛЕЙ FisFM32 И ДИСПЛЕЯ .....	8
3.3. УСТАНОВКА ЗАГЛУШКИ ДИСПЛЕЯ.....	9
3.4. ПОДКЛЮЧЕНИЕ ЯЩИКА ДЛЯ ДЕНЕГ .....	9
3.4.1. СИЛОВОЙ МОДУЛЬ BRIO VOX-FM32E .....	10
3.4.2. ВАРИАНТ ОТКРЫТИЯ ЯЩИКА КОМАНДАМИ ПРИНТЕРУ.....	10
3.4.3. ВАРИАНТ ОТКРЫТИЯ ЯЩИКА НЕПОСРЕДСТВЕННО POS-ом. ....	11
3.4.4. ВАРИАНТ ОТКРЫТИЯ ЯЩИКА ФИСКАЛЬНЫМ МОДУЛЕМ. ....	11
3.4.5. ВАРИАНТ ОТКРЫТИЯ ЯЩИКА ФИСКАЛЬНЫМ МОДУЛЕМ БЕЗ POS-а. ....	12
<b>4. ИНФОРМАЦИЯ ДЛЯ ПРОГРАММИСТОВ .....</b>	<b>13</b>
4.1. СОСТОЯНИЯ ФИСКАЛЬНОГО МОДУЛЯ.....	13
4.2. БИБЛИОТЕКА ФУНКЦИЙ ДЛЯ РАБОТЫ С BRIO FisFM32/32E .....	14
4.3. КОДЫ ОШИБОК .....	18
4.4. ДОСТУПНОСТЬ ФУНКЦИЙ .....	19
4.5. ПОСЛЕДОВАТЕЛЬНОСТЬ ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ ФУНКЦИЙ.....	20
4.6. ПРИМЕР ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ ФУНКЦИЙ (Delphi).....	20
<b>5. ЕС ДЕКЛАРАЦИЯ СООТВЕТСТВИЯ .....</b>	<b>22</b>

© BRIO EngineerinG. Авторские права защищены. BRIO EngineerinG®, BRIO, логотип BRIO EngineerinG, ShoppinG<sup>™</sup>, Fis-FM32/32E/32EM<sup>™</sup>, VOX-FM32E являются зарегистрированными торговыми марками фирмы SIA «BRIO SRC».

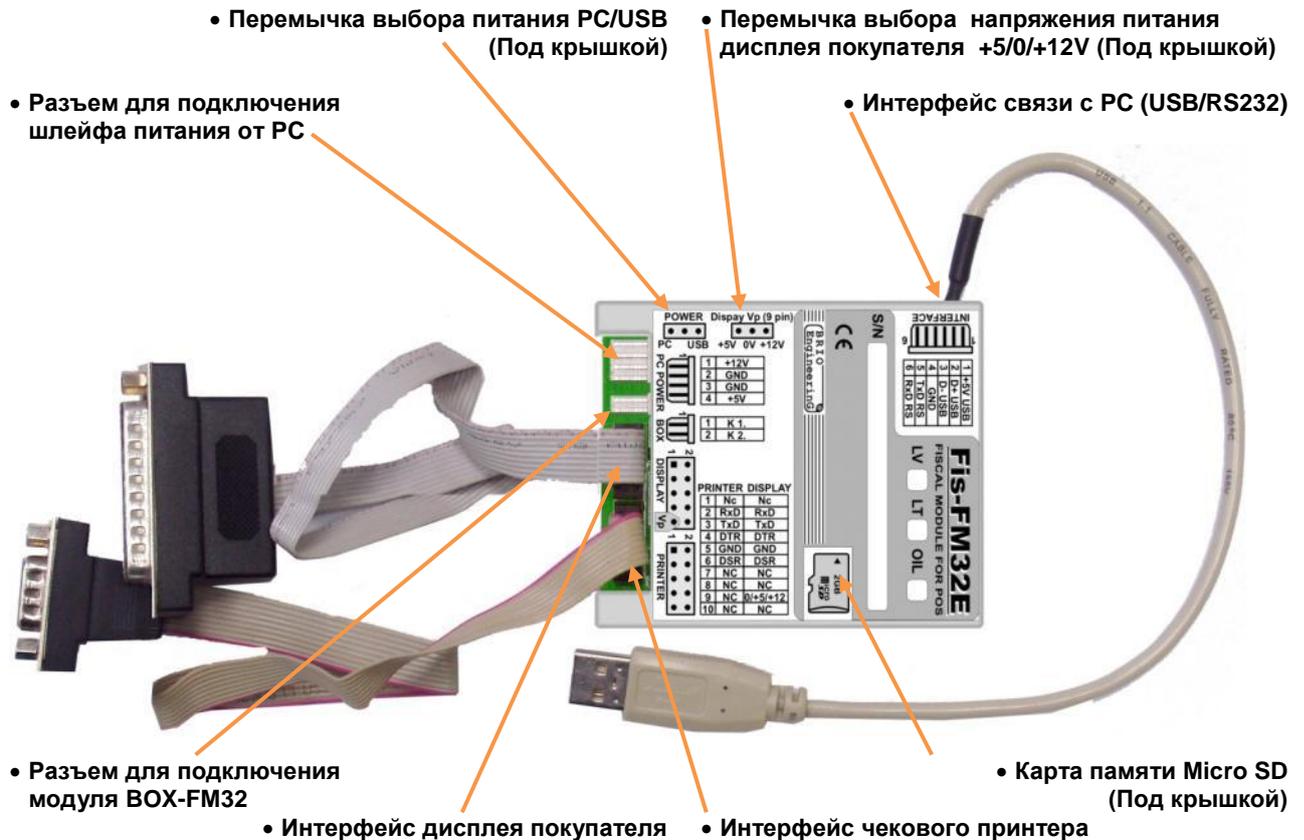
## 1. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

	Fis-FM32	Fis-FM32E	Fis-FM32EM
Интерфейс связи с PC, с автоматическим выбором типа интерфейса:	<ul style="list-style-type: none"> <li>• USB V2.0,</li> <li>• RS-232C (115200 Bits/sec. 8 Bits, NP, 1-Stop, Non flow control)</li> </ul>		<ul style="list-style-type: none"> <li>• USB V2.0,</li> </ul>
Тип встроенного процессора:	<ul style="list-style-type: none"> <li>• ARM</li> </ul>		
Максимальный ток потребления:	<ul style="list-style-type: none"> <li>• 200 mA (Max)</li> </ul>		
Объем энергонезависимой памяти для текущих данных:	<ul style="list-style-type: none"> <li>• 4 MByte</li> </ul>		
Объем фискальной памяти ROM:	<ul style="list-style-type: none"> <li>• 8 MByte</li> </ul>		
Тип памяти для электронной контрольной ленты:	<ul style="list-style-type: none"> <li>• Нет</li> </ul>	<ul style="list-style-type: none"> <li>• Micro SD</li> </ul>	
Режим обмена PC↔Fis-FM32XX:	<ul style="list-style-type: none"> <li>• Пакетный обмен по специальному протоколу.</li> </ul>		
Фискализация и инициализация:	<ul style="list-style-type: none"> <li>• Однократная</li> </ul>		
Максимальное количество Z-отчетов:	<ul style="list-style-type: none"> <li>• 8192</li> </ul>	<ul style="list-style-type: none"> <li>• 1800</li> </ul>	
Аппаратная диагностика:	<ul style="list-style-type: none"> <li>• Встроенная система диагностики ошибок</li> <li>• Встроенные часы и календарь</li> </ul>		
Защита от стирания:	<ul style="list-style-type: none"> <li>• Схема блокировки команды стирания.</li> </ul>		
Интерфейс принтера:	<ul style="list-style-type: none"> <li>• RS-232C</li> </ul>		
Типы поддерживаемых принтеров:  *** Для фискального модуля Fis-FM32E начиная с версии прошивки V5	ПРИНТЕРА С КОНТРОЛЬНОЙ ЛЕНТОЙ: <ul style="list-style-type: none"> <li>• Epson TM-U210A</li> <li>• Epson TM-U220A</li> <li>• Epson TM-U260A</li> <li>• Bixolon SRP-270D</li> <li>• + Любые принтера, совместимые с вышеперечисленными по системе команд и интерфейсу.</li> </ul>	ПРИНТЕРА С КОНТРОЛЬНОЙ ЛЕНТОЙ: <ul style="list-style-type: none"> <li>• Epson TM-U210A</li> <li>• Epson TM-U220A</li> <li>• Epson TM-U260A</li> <li>• Bixolon SRP-270D</li> </ul> ТЕРМОПРИНТЕРА: <ul style="list-style-type: none"> <li>• Bixolon SRP-350Plus</li> <li>• Partner Tech RP-300</li> <li>• Bixolon SRP-275C</li> <li>• Bixolon SRP-370</li> <li>• Bixolon SRP-500</li> <li>• Partner Tech PT-6200</li> <li>• + Любые принтера, совместимые с вышеперечисленными по системе команд и интерфейсу.</li> </ul>	
Возможность печати в заголовке чека одного из логотипов, предварительно загруженных в принтер:  *** Для фискальных модулей Fis-FM32E(M) начиная с версии внутренней прошивки V5.	<ul style="list-style-type: none"> <li>• Нет</li> </ul>	ПРИНТЕРА С ВОЗМОЖНОСТЬЮ ЗАГРУЗКИ ЛОГОТИПА: <ul style="list-style-type: none"> <li>• Bixolon SRP-350Plus</li> <li>• PartnerTech RP-320</li> <li>• Bixolon SRP-275</li> <li>• Epson TM-T260F</li> <li>• Epson TM-U220A</li> <li>• Bixolon SPR-270</li> <li>• + Любые принтера, совместимые с вышеперечисленными по системе команд и интерфейсу.</li> </ul>	

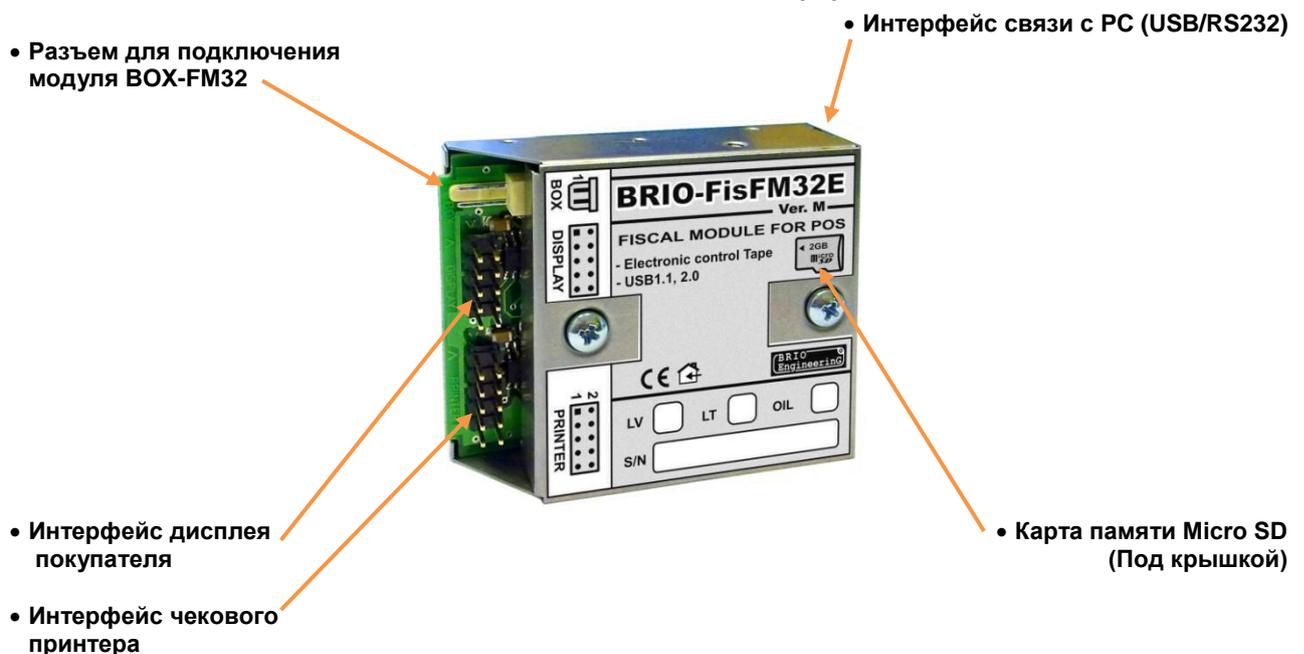
<b>Интерфейс дисплея покупателя:</b>	<ul style="list-style-type: none"> <li>• RS-232C</li> <li>• На 9-й вывод разъема можно подать напряжение +5V, или +12V для питания дисплея.</li> </ul>	<ul style="list-style-type: none"> <li>• RS-232C</li> </ul>
<b>Типы поддерживаемых дисплеев покупателя:</b>	<ul style="list-style-type: none"> <li>• CD7220</li> <li>• Epson</li> <li>• Vixolon</li> <li>• + Любые другие дисплеи, совместимые с вышеперечисленными по системе команд и интерфейсу.</li> </ul>	
<b>Управление ящиком для денег:</b>	<ul style="list-style-type: none"> <li>• Командами через чековый принтер.</li> </ul>	<ul style="list-style-type: none"> <li>• Командами через чековый принтер.</li> <li>• Непосредственно, через силовой модуль <b>BOX-FM32E</b></li> </ul>
<b>Габаритные размеры (mm):</b>	<ul style="list-style-type: none"> <li>• 88 x 73 x 19</li> </ul>	<ul style="list-style-type: none"> <li>• 60 x 50 x 19</li> </ul>

## 2. РАЗЪЕМЫ И ИНТЕРФЕЙСЫ

### 2.1. РАСПОЛОЖЕНИЕ РАЗЪЕМОВ BRIO-FisFM32, BRIO-FisFM32E



### 2.1. РАСПОЛОЖЕНИЕ РАЗЪЕМОВ BRIO-FisFM32E(M)



**2.2. ИНТЕРФЕЙС СВЯЗИ С КОМПЬЮТЕРОМ: RS-232C/USB**

Конт.	Сигнал	Направление	Назначение
1	DC +5V USB		Напряжения питания DC +5V
2	D + USB		Шина USB
3	D - USB		Шина USB
4	GND.L		Общий провод
5	TxD	Выход	Передача данных
6	RxD	Вход	Прием данных

**2.3. ИНТЕРФЕЙС ЧЕКОВОГО ПРИНТЕРА: RS-232C**

Конт.	Принтер	Назначение сигнала	КОНТАКТЫ НА РАЗЪЕМАХ ШЛЕЙФОВ	
			DB 25 (M)	DB 9 (M)
1	Nc	Не используется		
2	RxD	Прием данных	3	2
3	TxD	Передача данных	2	3
4	DTR	Запрос готовности	20	4
5	GND	Общий	7	5
6	DSR	Готовность принтера	6	6
7	Nc.	Не используется		
8	Nc.	Не используется		
9	Nc.	Не используется		

**2.1. ИНТЕРФЕЙС МОДУЛЯ ВОХ-FM32: ВОХ**

Конт.	Обознач.	Назначение сигнала
1	K1.	Контакт оптореле
2	K2.	Контакт оптореле

**2.1. ИНТЕРФЕЙС ДИСПЛЕЯ ПОКУПАТЕЛЯ: RS-232C**

Конт.	ДИСПЛЕЙ	НАЗНАЧЕНИЕ СИГНАЛА	КОНТАКТЫ НА РАЗЪЕМАХ ШЛЕЙФОВ	
			DB 25 (M)	DB 9 (M)
1	Nc	Не используется		
2	RxD	Прием данных	3	2
3	TxD	Передача данных	2	3
4	DTR	Запрос готовности	20	4
5	GND	Общий	7	5
6	DSR	Готовность дисплея	6	6
7	Nc	Не используется		
8	Nc	Не используется		
9	0/+5V/+12V	Питание дисплея покупателя	22	9
10	Nc	Не используется	Nc	Nc

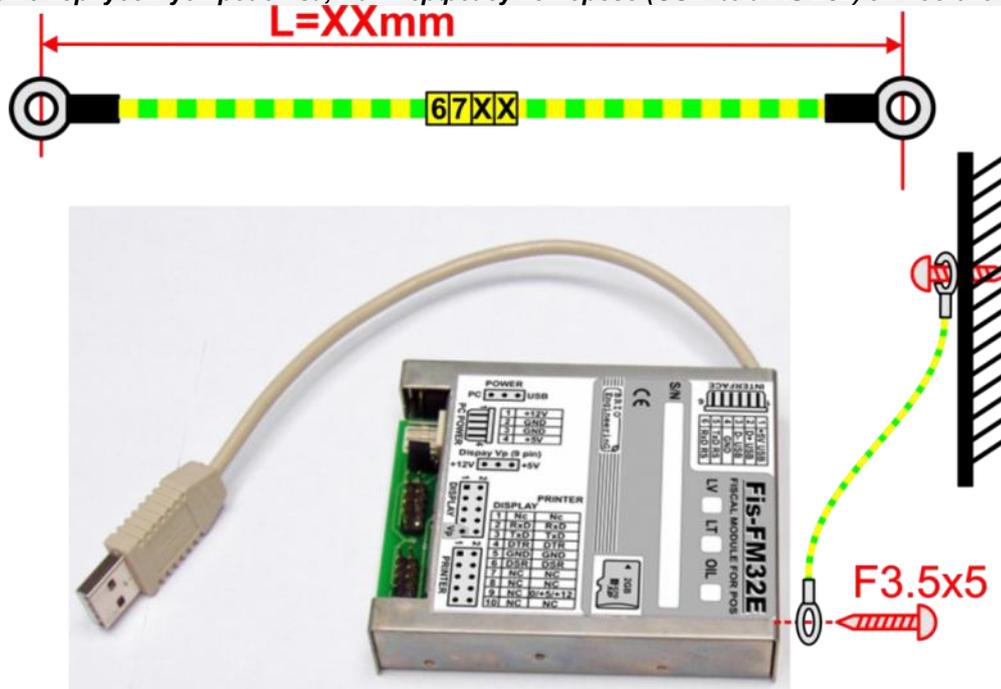
**2.1. РАЗЪЕМ ДЛЯ ПОДКЛЮЧЕНИЯ КАБЕЛЯ ПИТАНИЯ ОТ РС**

Конт.	Обознач.	НАЗНАЧЕНИЕ СИГНАЛА
1	+ 12V	DC +12V
2	GND	Общий провод
3	GND	Общий провод
4	+5V	DC +5V

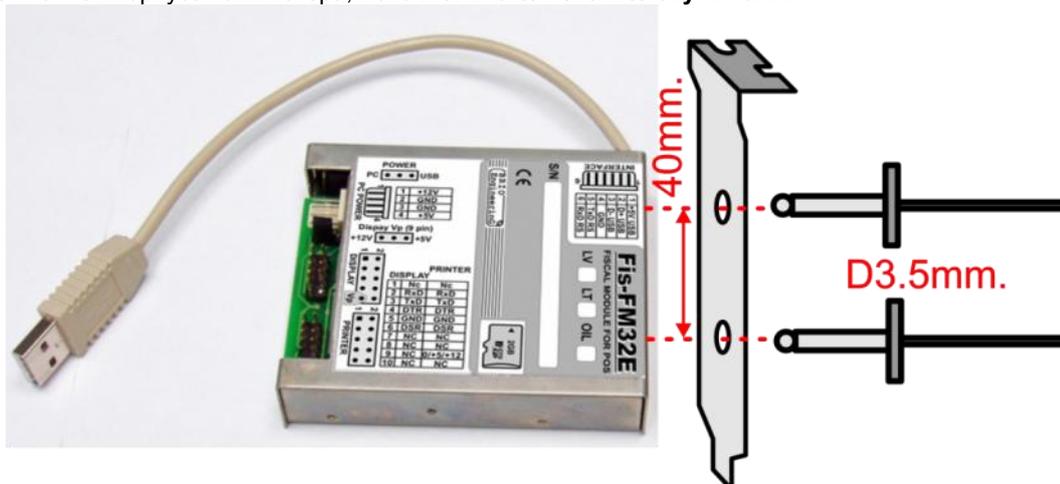
### 3. ПОДКЛЮЧЕНИЕ ФИСКАЛЬНОГО МОДУЛЯ

#### 3.1. МОНТАЖ И ЗАЗЕМЛЕНИЕ КОРПУСА

**ВНИМАНИЕ!!!** Металлический корпус фискального модуля должен быть электрически надежно соединен с корпусом устройства, к интерфейсу которого (USB или RS-232) он подключен!

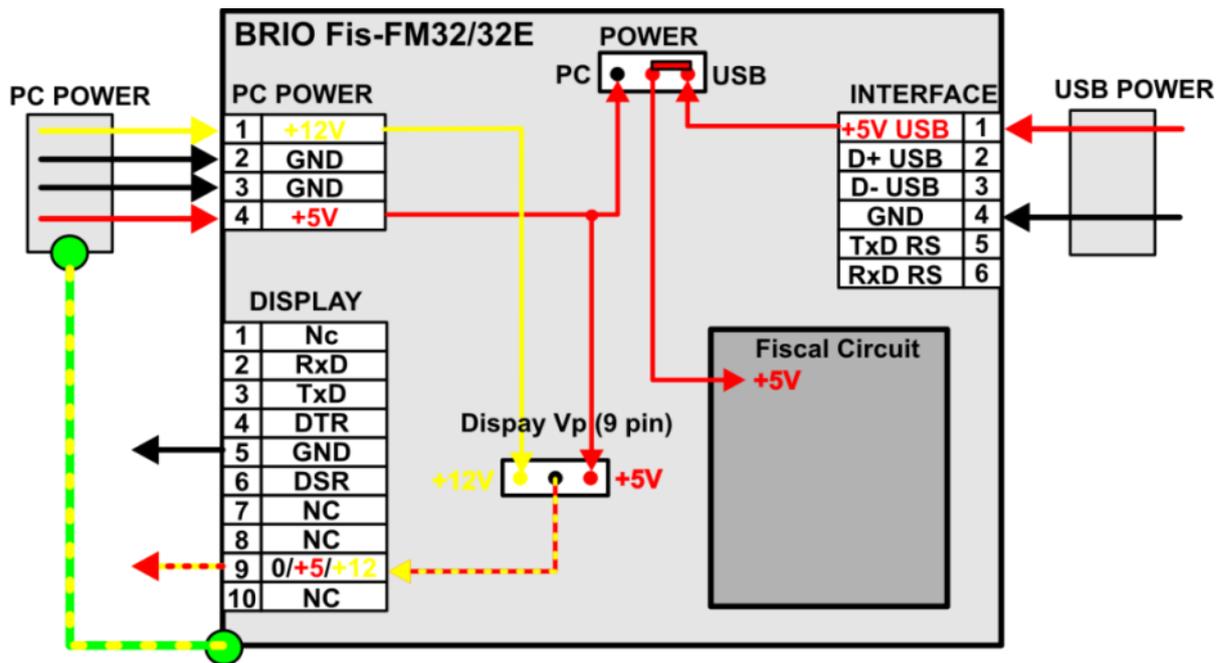


- Для обеспечения надежного заземления корпуса фискального модуля, следует использовать отдельный кабель **СВ-67XX**.
- Кабель необходимо закрепить саморезом F3.5x5 (Длиной не более 5mm!) в одно из отверстий на торцевой части корпуса модуля.
- Второй конец кабеля следует закрепить на металлической части, надежно связанной с общим заземлением корпуса компьютера, максимально близко к блоку питания.



- Фискальный модуль может быть установлен в компьютер при помощи стандартной планки. Планку следует предварительно прикрепить к корпусу фискального модуля при помощи двух вытяжных заклепок диаметром 3,5 мм.
- Если фискальный модуль установлен в корпусе компьютера при помощи стандартной планки, то дополнительного заземляющего кабеля не требуется.

### 3.2. ПИТАНИЕ ФИСКАЛЬНЫХ МОДУЛЕЙ FisFM32 И ДИСПЛЕЯ



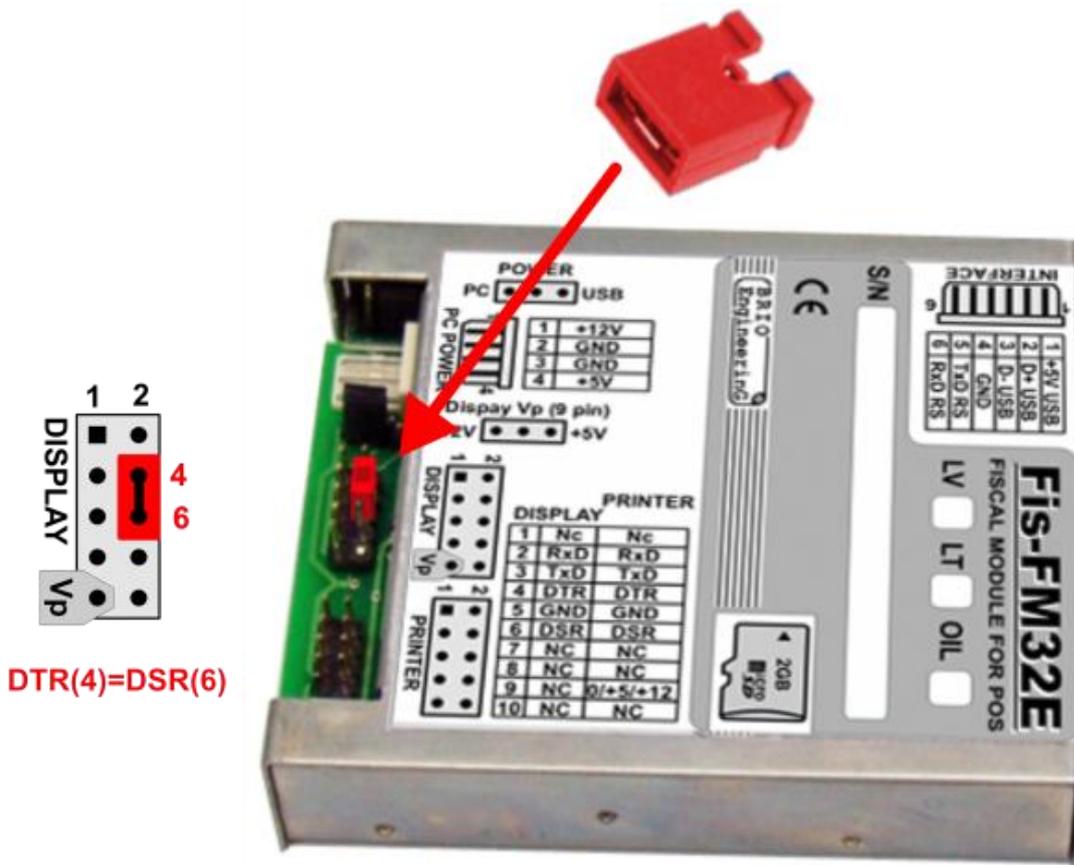
- Напряжение питания на ФМ может подаваться как разъема **INTERFACE**, так и с разъема **PC POWER**, что выбирается положением переключки **POWER**, расположенной внутри корпуса.
- Если для работы ФМ используется интерфейс USB, то питание следует подавать с интерфейсного разъема **INTERFACE** (Положение переключки **POWER** –USB).
- Если используется интерфейс RS-232, то к разъему PC POWER следует подключить стандартный кабель от блока питания компьютера, а переключку **POWER** установить в положение PC.

**ВНИМАНИЕ!!! Категорически запрещается при использовании интерфейса USB, использовать питание от разъема PC POWER (Положение переключки – PC)!!!**

- Для питания дисплея покупателя на (9) контакт разъема **DISLPAY** можно подать напряжение питания. Для этого к разъему **PC POWER** следует подключить стандартный кабель от блока питания компьютера, а переключку (Display Vp) установить в положение +5V, или +12V, в зависимости от величины требуемого напряжения.

### 3.3. УСТАНОВКА ЗАГЛУШКИ ДИСПЛЕЯ.

- Если к фискальному модулю дисплей подключать не требуется, то необходимо в обязательном порядке установить заглушку так, как это показано на рисунке.



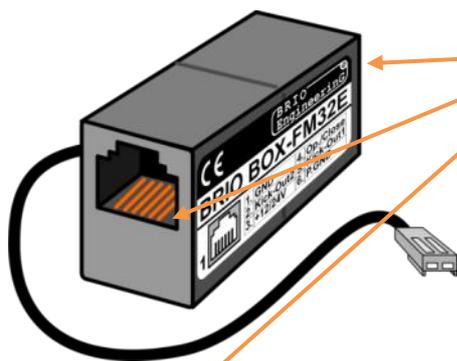
**ВНИМАНИЕ!!! При отсутствии Дисплея и Заглушки работа ФМ будет заблокирована.**

### 3.4. ПОДКЛЮЧЕНИЕ ЯЩИКА ДЛЯ ДЕНЕГ.

- Встроенная программа фискальных модулей позволяет открывать ящик для денег только в случаях, которые определены в текущем законодательстве. Причем делает это одновременно двумя способами:
  - В виде подачи отдельной команды чековому принтеру на открытие подключенного к нему денежного ящика.
  - Замыканием контактов К1 и К2 на разъеме ВОХ, к которому должен быть подключен силовой модуль ВОХ-FM32E/EM. (Только для версий фискального модуля, имеющих этот разъем.)

**ВНИМАНИЕ!!! Категорически запрещается использовать выход ВОХ фискального модуля для непосредственной коммутации электромагнита денежного ящика, без использования силового модуля BRIO ВОХ-FM32E.**

## 3.4.1. СИЛОВОЙ МОДУЛЬ BRIO BOX-FM32E



- Разъемы для подключения кабеля от ящика для денег, или кабелей для подачи питания.

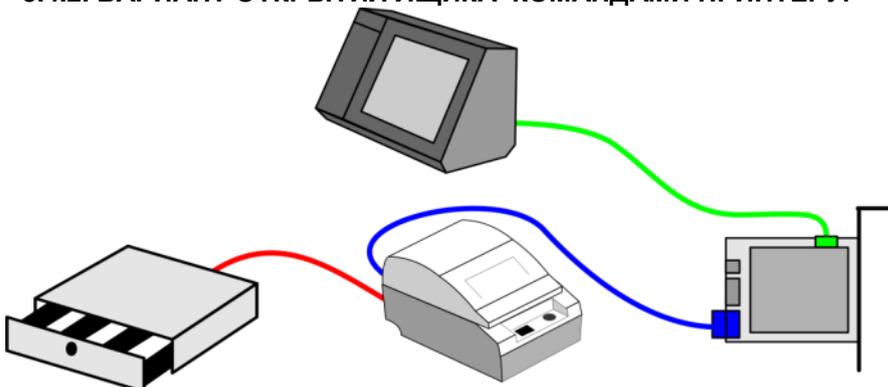
**ВНИМАНИЕ!!!** Все контакты одного из этих разъемов соединены с аналогичными контактами другого разъема. Поэтому разъемы - взаимозаменяемы.

- Кабель для подключения силового модуля к разъему BOX фискального модуля.



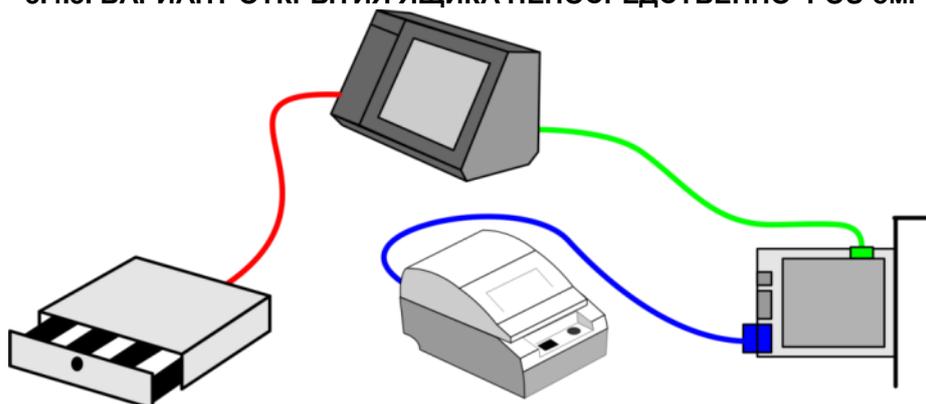
Конт.	Обозначение	НАЗНАЧЕНИЕ СИГНАЛА
1	GND	Общий провод
2	Kick Out-2	Ключ-2 электромагнита ящика (Не использ.)
3	DC +12/24V	Напряжение питания
4	Open/Close	Датчик открытия ящика
5	Kick Out-1	Ключ-1 электромагнита ящика
6	P.GND	Защитная земля

## 3.4.2. ВАРИАНТ ОТКРЫТИЯ ЯЩИКА КОМАНДАМИ ПРИНТЕРУ.



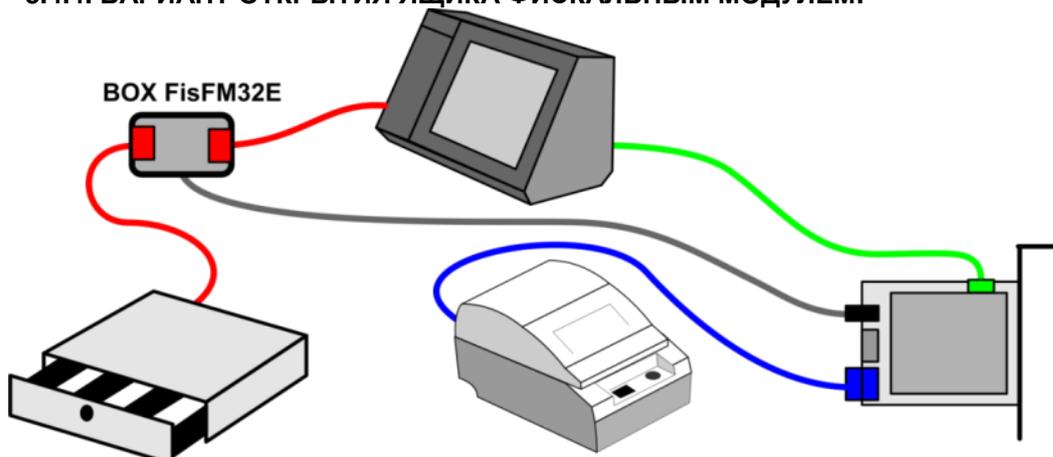
- Этот вариант управления является предпочтительным и применяется, если у чекового принтера есть специальный интерфейс для подключения ящика для денег.
- Ящик для денег подключен к соответствующему разъему чекового принтера.
- ФМ в нужный момент подает команду на открытие ящика, и принтер открывает ящик для денег.

### 3.4.3. ВАРИАНТ ОТКРЫТИЯ ЯЩИКА НЕПОСРЕДСТВЕННО POS-ом.



- Этот вариант управления применяется, если у чекового принтера нет интерфейса для подключения ящика для денег, а в POS-е он имеется. Кроме того, такой вариант можно применять тогда, когда требуется открывать ящик не только в случаях, оговоренных законодательством.
- Ящик для денег подключен к соответствующему разъему POS-а. (Имеется не у всех моделей)
- Открытие ящика для денег происходит непосредственно командами программного обеспечения POS-а. (Такая возможность имеется не у всех версий программного обеспечения)

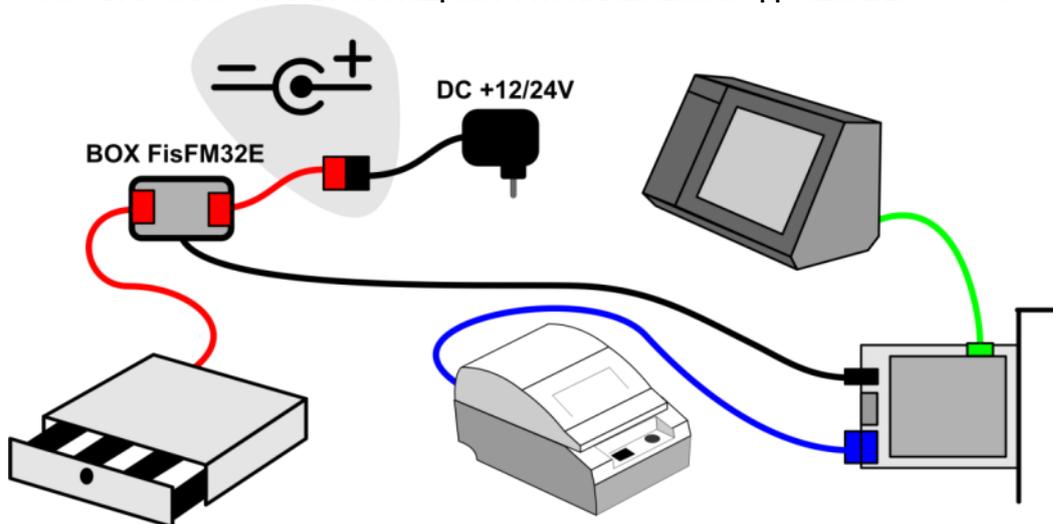
### 3.4.4. ВАРИАНТ ОТКРЫТИЯ ЯЩИКА ФИСКАЛЬНЫМ МОДУЛЕМ.



- Этот вариант управления применяется, если у чекового принтера нет интерфейса для подключения ящика для денег, в POS-е такой интерфейс имеется, но программное обеспечение POS-а не поддерживает открытие ящика.
- Ящик для денег подключен к силовому модулю BRIO BOX-FM32E, а он, в свою очередь к соответствующему разъему POS-а кабелем **CB-70XX** минимальной длины.
- Открытие ящика для денег происходит непосредственно через интерфейс BOX самим фискальным модулем.

**ВНИМАНИЕ!!!** При таком подключении ящик для денег может открываться двояко - как фискальным модулем, так и через интерфейс POS-а, если в дальнейшем на нем установить программное обеспечение, позволяющее управлять открытием ящика.

## 3.4.5. ВАРИАНТ ОТКРЫТИЯ ЯЩИКА ФИСКАЛЬНЫМ МОДУЛЕМ БЕЗ POS-а.



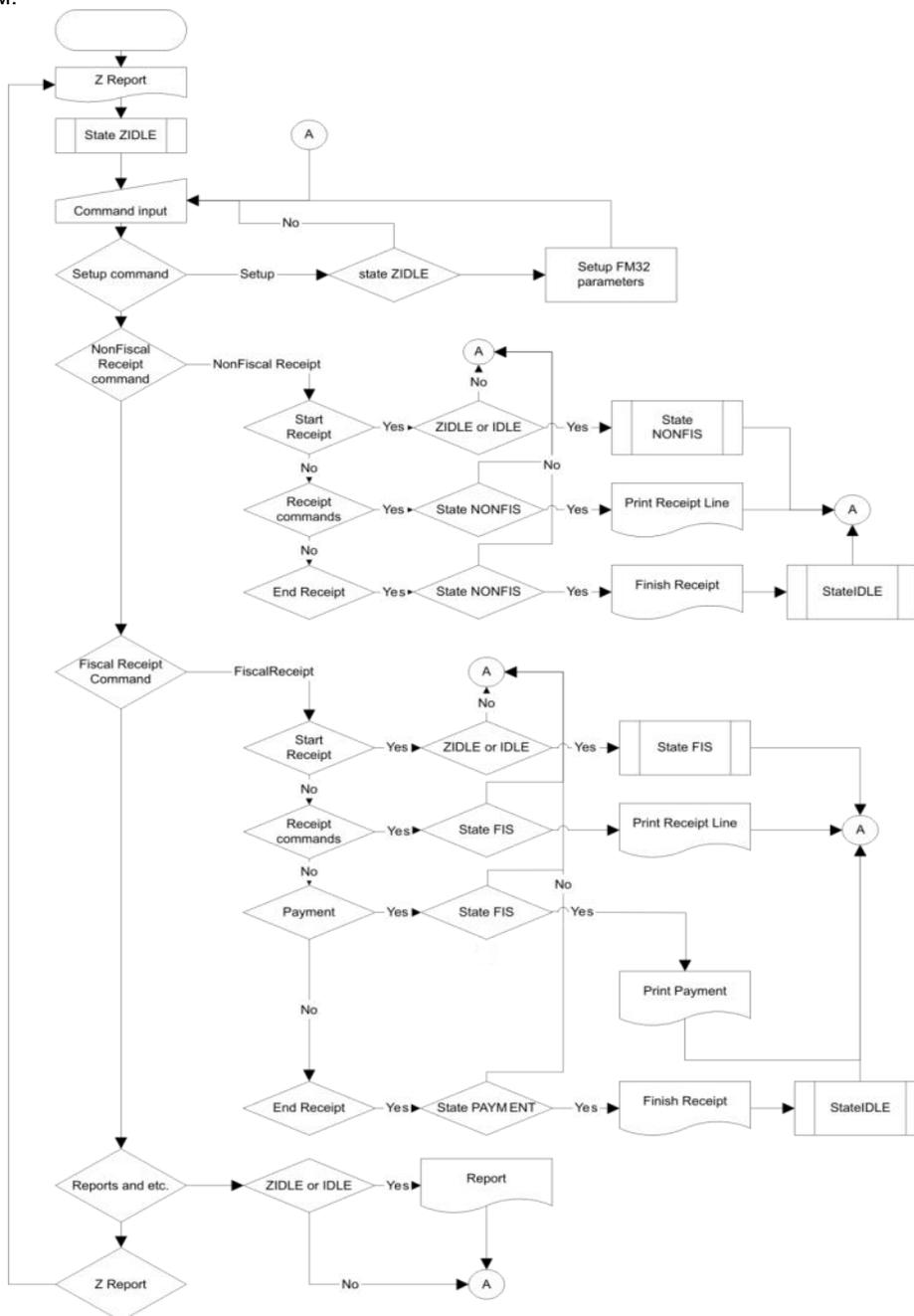
- Этот вариант управления применяется, если у чекового принтера и POS-а нет интерфейса для подключения ящика для денег.
- Ящик для денег подключен к силовому модулю BRIO BOX-FM32E, к которому при помощи кабеля **CB-71XX** минимальной длины и соблюдая полярность, подключен внешний блок питания.
- Блок питания должен иметь выходное напряжение, требуемое для данной модели денежного ящика (Обычно DC +12, или +24V), и максимально допустимый ток 500- 800 mA.
- Открытие ящика для денег происходит непосредственно через интерфейс BOX самим фискальным модулем.

## 4. ИНФОРМАЦИЯ ДЛЯ ПРОГРАММИСТОВ

### 4.1. СОСТОЯНИЯ ФИСКАЛЬНОГО МОДУЛЯ.

Фискальный модуль может находиться в пяти разных состояниях. Команды могут выполняться только в тех состояниях, для которых они разрешены. После выполнения некоторых команд, фискальный модуль может менять свое состояние.

- **IDLE** - Фискальный модуль переходит в это состояние, после распечатки очередного чека.
- **ZIDLE** - Фискальный модуль переходит в это состояние, после выполнения Z-отчета.
- **FIS** - Состояние фискального модуля, когда начато оформление нового фискального чека, но еще не оформлена его оплата.
- **NONFIS** - Состояние, когда начат, но еще не завершен нефискальный чек.
- **HARD** - Фискальный модуль переходит в это состояние в случае ошибки при работе с дисплеем, или чековым принтером.



АЛГОРИТМ ПРОХОЖДЕНИЯ КОМАНД

## 4.2. БИБЛИОТЕКА ФУНКЦИЙ ДЛЯ РАБОТЫ С BRIO FisFM32/32E

### СБРОС СОСТОЯНИЯ ФИСКАЛЬНОГО МОДУЛЯ

int ResetFiscal( void )

### УСТАНОВКА ТИПА ЧЕКОВОГО ПРИНТЕРА

int SetPrinterType( int printerType )

Используемые типы принтеров:

- printerEpson210 = 0
- printerAllThermalPrinters = 10,
- printerSRP275 = 11

### УСТАНОВКА ТИПА ЧЕКОВОГО ПРИНТЕРА С УКАЗАНИЕМ ВИДА КОДИРОВКИ СИМВОЛОВ

int SetPrinterTypeEx( int printerType, int coding )

Виды кодировки символов:

- codingDOS = 0
- codingWin = 1

### УСТАНОВКА ТИПА ЧЕКОВОГО ПРИНТЕРА С УКАЗАНИЕМ ВИДА КОДИРОВКИ СИМВОЛОВ И НАСТРОЙКАМИ ДЛЯ ПЕЧАТИ ЛОГОТИПА

int SetPrinterTypeEx2( int printerType, int coding, int image, int imagenumber, int imagelarge );

- printerType - из функции SetPrinterType
- coding - из функции SetPrinterTypeEx
- image – 1- использовать, 0 – не использовать загруженный в принтер логотип
- imagenumber – Порядковый номер загруженного в принтер логотипа 1..8
- imagelarge – 1- удваивать, 0 – не удваивать размер логотипа при печати

### УСТАНОВКА ИНТЕРВАЛА МЕЖДУ СТРОКАМИ НА ЧЕКОВОМ ПРИНТЕРЕ

int SetCompressionMode( int compression )

### УСТАНОВКА ДЛИТЕЛЬНОСТИ ИМПУЛЬСОВ ДЛЯ ОТКРЫВАНИЯ ДЕНЕЖНОГО ЯЩИКА

int SetTillImpuls( int time1, int time2 )

### УСТАНОВКА ДАТЫ ФИСКАЛЬНОГО МОДУЛЯ

int SetDate( char \* dateString )

Дата в формате "YYYY.MM.DD"

### УСТАНОВКА ВРЕМЕНИ ФИСКАЛЬНОГО МОДУЛЯ

int SetTime( char \* timeString )

Время в формате "HH:MM"

### УСТАНОВКА ЗАГОЛОВКА ЧЕКА

int SetHeader( char \* line1, char \* line2, char \* line3, char \* line4 )

### УСТАНОВКА ЗАГОЛОВКА ЧЕКА С УКАЗАНИЕМ ПАРАМЕТРОВ ФОРМАТИРОВАНИЯ СТРОК

int SetHeaderEx( int attr1, char \* line1, int attr2, char \* line2, int attr3, char \* line3, int attr4, char \* line4 )

### УСТАНОВКА СТРОК ОКОНЧАНИЯ ЧЕКА С УКАЗАНИЕМ ПАРАМЕТРОВ ФОРМАТИРОВАНИЯ СТРОК

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

### РАЗРЕШЕНИЕ/ЗАПРЕЩЕНИЕ РАСПЕЧАТКИ СТРОК ОКОНЧАНИЯ ЧЕКА

int EnableFooter( int enable )

### УСТАНОВКА ДОПОЛНИТЕЛЬНЫХ СТРОК ОКОНЧАНИЯ ЧЕКА С ФОРМАТИРОВАНИЕМ

int SetFooter2( int attr1, char \* line1, int attr2, char \* line2 )

### РАЗРЕШЕНИЕ/ЗАПРЕЩЕНИЕ РАСПЕЧАТКИ ДОПОЛНИТЕЛЬНЫХ СТРОК ОКОНЧАНИЯ ЧЕКА

int EnableFooter2( int enable )

### УСТАНОВКА СТАВКИ НАЛОГА

int SetVAT( int number, double rate )

int SetVat( int number, double rate )

### УСТАНОВКА ПАРАМЕТРОВ ВАЛЮТЫ

int SetCurrency( int number, char \* name, double rate )

**УСТАНОВКА НАИМЕНОВАНИЯ БЕЗНАЛИЧНОГО ВИДА РАСЧЕТА**

int SetCredit( int number, char \* name )

**РАЗРЕШЕНИЕ/ЗАПРЕЩЕНИЕ ОПЕРАЦИИ ВОЗВРАТА**

int AllowGoodsReturn( char \* unitNumber )

**ПОЛУЧЕНИЕ ИНФОРМАЦИИ ИЗ ФИСКАЛЬНОГО МОДУЛЯ**

int GetFiscalInfo( int infoType, char \* data )

- infoReceiptSumm = 0 – общая сумма начатого чека
- infoShiftTurnover = 1 – общий оборот за смену
- infoReceiptNumber = 2 – текущий номер чека
- infoUnitNumber = 3 – номер фискального блока
- intUnitVersion = 4 – версия фискального блока
- infoDate = 5 – дата в фискальном блоке
- infoTime = 6 – время в фискальном блоке
- infoReportNumber = 7 – текущий номер Z-отчета
- infoDayReceiptNumber = 8 – номер фискального чека за смену
- infoUnitState = 9 – состояние фискального модуля
- infoHeader = 10 – строки заголовка чека
- infoFooter = 11 – строки окончания чека
- infoFooter2 = 12 – дополнительные строки окончания чека
- infoFooterEnabled = 13 – разрешение печати строк окончания чека
- infoFooter2Enabled = 14 – разрешение печати дополнительных строк окончания чека
- infoCurDescription = 30 – элементы таблицы валют
- infoTax = 40 – элементы таблицы налогов
- infoCurCash = 50 – элементы таблицы остатка денег в денежном ящике
- infoCreditDescription = 60 – элементы таблицы наименований безналичных видов расчета
- infoTaxTurnover = 70 – элементы таблицы оборотов по видам налога
- infoTaxSumm = 80 – элементы таблицы сумм налогов по видам налога

**ВВОД РАЗМЕННЫХ ДЕНЕГ**

int MoneyIn( double money )

**ВВОД РАЗМЕННЫХ ДЕНЕГ В ВАЛЮТЕ**

int MoneyInCurr( int number, double money );

// Extract money from cash drawer

**ИНКАССАЦИЯ**

int MoneyOut( double money )

**ИНКАССАЦИЯ В ВАЛЮТЕ**

int MoneyOutCurr( int number, double money )

int MoneyOutCurr2( int number, double money )

**ОТКРЫТЬ ДЕНЕЖНЫЙ ЯЩИК**

int OpenCashDrawer( void )

int OpenCachDrower( void )

**РАСПЕЧАТАТЬ ТАБЛИЦУ НАЛОГОВ НА ЧЕКОВОМ ПРИНТЕРЕ**

int PrintVATTable( void )

int PrintVatTable( void )

**РАСПЕЧАТАТЬ ТАБЛИЦУ ВАЛЮТ НА ЧЕКОВОМ ПРИНТЕРЕ**

int PrintCurrencyTable( void )

**РАСПЕЧАТАТЬ ТАБЛИЦУ НАИМЕНОВАНИЙ БЕЗНАЛИЧНЫХ ВИДОВ ОПЛАТЫ НА ЧЕКОВОМ ПРИНТЕРЕ**

int PrintCreditTable( void )

**РАСПЕЧАТАТЬ Z-ОТЧЕТ**

int PrintZReport( void )

**РАСПЕЧАТАТЬ X-ОТЧЕТ**

int PrintXReport( void )

**РАСПЕЧАТАТЬ МИНИ X-ОТЧЕТ**

int PrintMiniXReport( void )

**РАСПЕЧАТАТЬ СУММАРНЫЙ ПЕРИОДИЧЕСКИЙ ОТЧЕТ ПО ДИАПАЗОНУ ДАТ**

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

Даты в формате "YYYY.MM.DD".

**РАСПЕЧАТАТЬ СУММАРНЫЙ ПЕРИОДИЧЕСКИЙ ОТЧЕТ ПО ДИАПАЗОНУ НОМЕРОВ**

int PrintSumPeriodicReportByNumber( int number1, int number2 )

**РАСПЕЧАТАТЬ ПЕРИОДИЧЕСКИЙ ОТЧЕТ ПО ДИАПАЗОНУ ДАТ**

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

Даты в формате "YYYY.MM.DD".

**РАСПЕЧАТАТЬ ПЕРИОДИЧЕСКИЙ ОТЧЕТ ПО ДИАПАЗОНУ НОМЕРОВ**

int PrintPeriodicReportByNumber( int number1, int number2 )

**ВЫВЕСТИ ИНФОРМАЦИЮ НА ДИСПЛЕЙ ПОКУПАТЕЛЯ**

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

int CustomerDisplay2( char \* line1, char \* line2 )

int CustomerDisplayPro( char \* command )

displayType – тип дисплея : 1

**НАЧАТЬ НЕФИСКАЛЬНЫЙ ЧЕК**

int BeginNonFiscalReceipt( void )

**ПЕЧАТАТЬ СТРОКУ С ТАРОЙ**

int PrintTareItem( char \* name, double quantity, double price )

**ОТКАЗ ОТ ПОЗИЦИИ ТАРЫ**

int PrintTareItemVoid( char \* name, double quantity, double price )

**ПЕЧАТАТЬ СТРОКУ С ДЕПОЗИТОМ**

int PrintDepositReceive( char \* name, double quantity, double price )

**ОТКАЗ ОТ ПОЗИЦИИ ДЕПОЗИТА**

int PrintDepositRefund( char \* name, double quantity, double price )

**ПЕЧАТЬ ИНФОРМАЦИОННОЙ СТРОКИ В НЕФИСКАЛЬНОМ ЧЕКЕ**

int PrintNonFiscalLine( char \* line, int attribute )

**ЗАВЕРШИТЬ НЕФИСКАЛЬНЫЙ ЧЕК**

int EndNonFiscalReceipt( void )

**НАЧАТЬ ФИСКАЛЬНЫЙ ЧЕК**

int BeginFiscalReceipt( void )

**ПЕЧАТЬ СТРОКИ ФИСКАЛЬНОГО ЧЕКА**

int PrintRecltem( char \* name, double quantity, double price, int taxNumber, char \* unit )

**ОТКАЗ ОТ СТРОКИ ФИСКАЛЬНОГО ЧЕКА**

int ItemReturn( char \* name, double quantity, double price,  
int taxNumber, char \* unit, int depart,  
double discountPercent, double discountSumm )

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

**ПЕЧАТЬ СТРОКИ КОММЕНТАРИЯ В ФИСКАЛЬНОМ ЧЕКЕ**

int PrintCommentLine( char \* line, int attribute )

**СКИДКА НА ПОЗИЦИЮ ЧЕКА**

int DiscountAdditionForItem( int type, double val )

Тип скидки:

- dtPcnt = 1 – в процентах
- dtSumm = 2 – абсолютное значение

**СКИДКА НА ЧЕК**

int DiscountAdditionForReceipt( int type, double val )

Тип скидки:

- dtPcnt = 1 – в процентах
- dtSumm = 2 – абсолютное значение

#### **ЗАВЕРШИТЬ ОФОРМЛЕНИЕ ФИСКАЛЬНЫЙ ЧЕК**

```
int EndFiscalReceipt( double summCash, double summCredit1, double summCredit2,
                    double summCredi3, double summCredi4 )
```

```
int EndFiscalReceiptCurr( double summCash, double summCredit1,
                        double summCredit2, double summCredit3,
                        double summCredit4, double summCur1, double summCur2,
                        double summCur3 )
```

```
int 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 )
```

#### **ЗАВЕРШИТЬ ОФОРМЛЕНИЕ ЧЕКА ВОЗВРАТА**

```
int GoodsReturn( double summCash, double summCredit1,
                double summCredit2, double summCredit3,
                double summCredit4 )
```

```
int GoodsReturnCurr( double summCash, double summCredit1,
                    double summCredit2, double summCredit3,
                    double summCredit4,
                    double summCur1, double summCur2,
                    double SummCur3 )
```

```
int 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 )
```

#### **ПЕЧАТЬ КОПИИ ЧЕКА**

```
int PrintCopyOfLastReceipt( void )
```

#### **ПЕЧАТЬ СООБЩЕНИЯ НА ЧЕКОВОМ ПРИНТЕРЕ**

```
int PrintErrorMessage( char * message )
```

#### **ФИСКАЛИЗАЦИЯ МОДУЛЯ**

```
int Fiscalization( char * date )
```

#### **ПРОГРАММИРОВАНИЕ РЕГИСТРАЦИОННОГО НОМЕРА (ТОЛЬКО ДЛЯ ЛИТОВСКОЙ ВЕРСИИ )**

```
int SetCompanyVATCode( char regNum )
```

#### **ПРОГРАММИРОВАНИЕ НАЗВАНИЕ ОТДЕЛА**

```
int SetDepartName( int number, char * name )
```

#### **ПОЛУЧЕНИЕ ДАТЫ ИЗ ФИСКАЛЬНОГО МОДУЛЯ**

```
int GetFiscalData( int infoType, char data[10] )
```

#### **ЗАПИСЬ ИДЕНТИФИКАЦИОННОГО НОМЕРА СИСТЕМЫ**

```
int SetId( char * id )
```

#### **ЧТЕНИЕ ИДЕНТИФИКАЦИОННОГО НОМЕРА СИСТЕМЫ**

```
int GetIdNumber( char * date )
```

#### **ЧТЕНИЕ НОМЕРА ШАССИ**

```
int GetUnitNumber( char * date )
```

#### **КОРРЕКЦИЯ ТОВАРА В ЧЕКЕ**

```
int ItemReturnDepart( char * name, double quantity, double price,
                    int taxNumber, char * unit, int depart,
                    double discountPercent, double discountSumm )
```

**ПРОДАЖА ТОВАРА В ЧЕКЕ**

int PrintRecltemDepart( char \* name, double quantity, double price,  
int taxNumber, char \* unit, int depart )

**ЗАПИСЬ НОМЕРА ШАССИ**

int SetShassi( char \* number )

**4.3. КОДЫ ОШИБОК**

Fis-FM32/32E ERRORS	ЗНАЧЕНИЕ		ПРИМЕЧАНИЕ
	DEC	HEX	
FM32_OK	0	0x00	Ошибок нет.
ERR_LENGTH	4	0x04	Недопустимая длина пакета данных.
ERR_DATA	5	0x05	Недопустимые данные в пакете.
ERR_XOR	6	0x06	Ошибочная контрольная сумма пакета данных.
ERR_ETX	7	0x07	Отсутствует символ конца пакета.
ERR_ILLEGAL	16	0x10	Ошибочная или неподдерживаемая команда.
ERR_IDLE_STATE	17	0x11	Команда не может быть выполнена, так как FM32 находится в состоянии IDLE.
ERR_NONFIS_STATE	18	0x12	Команда не может быть выполнена, так как FM32 находится в состоянии NONFIS.
ERR_FIS_STATE	19	0x13	Команда не может быть выполнена, так как FM32 находится в состоянии FIS.
ERR_HARD_STATE	20	0x14	Ошибка работы с внешними устройствами. Например принтером, или дисплеем покупателя.
ERR_PARAMETERS	21	0x15	Ошибочные параметры в пакете. Недопустимые параметры для команды или параметры содержат недопустимые значения.
ERR_ITEM_DESC_LENGTH	22	0x16	Длина параметра в команде превышает допустимое.
ERR_ITEM_OUANTITY	23	0x17	Недопустимое количество.
ERR_ITEM_PRICE	24	0x18	Недопустимая цена.
ERR_VAT	25	0x19	Недопустимый номер налога. Номера могут быть в диапазоне от 0 до 4.
ERR_ITEM_DIM	26	0x1A	Недопустимое название единицы измерения. Название не может содержать более 4 символов.
ERR_DEFICIENT_PAYMENT	27	0x1B	Полученные от покупателя деньги меньше, чем сумма покупки.
ERR_OVERPAYMENT_CREDIT	28	0x1C	Сумма безналичной оплаты превышает общую сумму по чеку.
ERR_ITEM_DISCOUNT	29	0x1D	Недопустимая скидка/наценка на товар. Возможные причины: процент скидки превышает 100% и т.д.
ERR_DISCOUNT_TYPE	30	0x1E	Недопустимый тип скидки/наценки.
ERR_COMMENT_LENGTH	31	0x1F	Недопустимая длина строки комментария.
ERR_PRINTER	32	0x20	Ошибка принтера.
ERR_DISPLAY	33	0x21	Ошибка дисплея покупателя.
ERR_PRICE_AMOUNT_OVERFLOW	34	0x22	Цена или количество товара больше чем 99999.99
ERR_FLASH_WRITE	35	0x23	Ошибка записи в фискальную память FM32.
ERR_NOT_FISCAL	36	0x24	FM32 не фискализован. Запрещены некоторые отчеты.
ERR_BAD_DATE	37	0x25	Ошибочный формат даты.
ERR_REPORT_NOT_FOUND	38	0x26	В указанном интервале дат отчеты отсутствуют.
ERR_FLASH_ERASE	39	0x27	Фискальная память FM32 недоступна для стирания или записи.
ERR_DISCOUNT_RECEIPT	40	0x28	Недопустимая скидка/наценка на товар.
ERR_NO_ITEMS	41	0x29	Скидка или наценка не привязаны к конкретному товару.
ERR_CANT_RETURN	42	0x2A	Попытка коррекция продажи еще не проданного товара.
ERR_OVER_ADDITION_PERCENT	43	0x2B	Процент наценки слишком большой
ERR_OVER_DISCOUNT_PERCENT	44	0x2C	Процент скидки слишком большой
ERR_OVER_ADDITION_FIXED	45	0x2D	Абсолютная наценка слишком большая.
ERR_OVER_DISCOUNT_FIXED	46	0x2E	Абсолютная скидка слишком большая.
ERR_NO_MONEY_FOR_DATA	47	0x2F	Недостаточно наличных денег в денежном ящике
ERR_ZERO_TOTAL	48	0x30	Сумма чека = 0

ERR_PAYMENT_NOT_EQUAL	49	0x31	Сумма возврата превышает общую сумму по чекам.
ERR_DEFICIENT_CASH_DRAWER	50	0x32	Недостаточно наличных денег в денежном ящике.
ERR_UNIQUE_FM32_NUMBER	51	0x33	Ошибочный серийный номер FM32.
ERR_NOTALLOW_GOODS_RETURN	52	0x34	Возврат товаров запрещен.
ERR_ALREADYFISCAL	54	0x36	FM32 уже фискализован.
ERR_NOT_SL_PRN	55	0x37	
ERR_RECEIPT_AMOUNT_OVER FLOW	56	0x38	Сумма по чеку превышает 9900000.
ERR_FLASH_FULL	57	0x39	Фискальная память FM32 полностью заполнена.
ERR_YEAR_VALUE	58	0x3A	Недопустимое значение года в дате
ERR_MONTH_VALUE	59	0x3B	Недопустимое значение месяца в дате
ERR_DAY_VALUE	60	0x3C	Недопустимое значение дня в дате.
ERR_JCLOSE_STATE	63	0x3F	После Z-отчета не считана контрольная лента
ERR_PAY_STATE	64	0x40	
ERR_CURRENCY_NUMBER	68	0x44	Недопустимый номер валюты
ERR_CURRENCY_RATE	69	0x45	Недопустимый курс валюты
ERR_CURRENCY_NOT_SET	70	0x46	Валюта не установлена.
ERR_CREDIT_OVERFLOW	72	0x48	Сумма безналичной оплаты превышает сумму по чекам.
ERR_TARE_QUANTITY	73	0x49	Недопустимое количество тары
ERR_CREDIT_ID	76	0x4C	Недопустимое ID безналичной оплаты
ERR_CREDIT_DESC	77	0x4D	Недопустимое название безналичной оплаты.
ERR_REFUND_IN_CURRENCY	78	0x4E	Выдача денег за возвращенный товар возможна только в базовой валюте.
ERR_CREDIT_NOT_SET	79	0x4F	Безналичная оплата не определена.
ERR_VAT_AMOUNT	80	0x50	Продажи с данным налогом отсутствуют.
ERR_OVER_BUF	100	0x64	Переполнения буфера товаров текущего чека.
ERR_DEPART_ID	128	0x80	Несуществующий номер отдела
ERR_DEPART_DESC	129	0x81	Неправильное название отдела
ERR_DEPART_SET	130	0x82	Ошибка предустановки параметров отдела
<b>Только для фискального модуля Fis-FM32E:</b>			
Err_NOFILE	131	0x83	Файл не найден
Err_DELFIL	132	0x84	Невозможно удалить файл
Err_FILE3YEAR	133	0x85	Удаление файла невозможно, так как не прошло еще три года с момента его создания.

FiscalUnit.DLL ERRORS	ЗНАЧЕНИЕ	ПРИМЕЧАНИЕ
ErrorDLLIllegalPacketStructure	-1	Неверная структура полей пакетов
ErrorDLLExecuteCmd	-2	Внутренняя ошибка при выполнении команды
ErrorUnknownCommand	16	Неизвестная команда
ErrorDidNotSendPacket	160	Невозможно отправить пакет
ErrorDidNotReceivePacket	161	Невозможно принять пакет
ErrorBadJournal	162	Неверная контрольная сумма электронной ленты

#### 4.4. ДОСТУПНОСТЬ ФУНКЦИЙ

Для правильной работы фискала и предотвращения ошибочных действий, в различных состояниях фискала доступны различные функции FiscalUnit.dll.

Возможные состояния фискала описаны в пункте «СОСТОЯНИЯ ФИСКАЛЬНОГО МОДУЛЯ». При вызове функции в не соответствующем состоянии возможно появление ошибок с номером 17,18,19 (см. «Коды ошибок»).

Для возврата в состояние IDLE или при возникновении ошибки нужно вызвать функцию ResetFiscal().

## 4.5. ПОСЛЕДОВАТЕЛЬНОСТЬ ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ ФУНКЦИЙ

Все функции, за исключением относящихся к чеку (как фискальному, так и не фискальному), возможно использовать в режиме IDLE. Для функций, относящихся к чеку, типичная последовательность вызова функций можно описать следующим образом:

- Вызов функции «начала чека». При этом меняется состояние фискала.
- Вызов функций «тело чека»
- Вызов функции «окончания чека». Состояние фискала возвращается в IDLE

## 4.6. ПРИМЕР ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ ФУНКЦИЙ (Delphi)

### ЗАГРУЗКА БИБЛИОТЕКИ:

```
procedure TTraceForm.FormCreate(Sender: TObject);
begin
  {You can use FiscalUnit.dll for local connect to FM32 or NetFiscalUnit.dll for using FM32 remote
  }
  // if IUse_Fm32Net then
  //   FisHandle := LoadLibrary('NetFiscalUnit.dll')
  // else
  FisHandle := LoadLibrary('FiscalUnit.dll');

  if FisHandle>0 then
    nFatalError:=0
  else
    nFatalError:=999;
end;
```

### ПЕЧАТЬ ФИСКАЛЬНОГО ЧЕКА:

```
//IReturn - Флаг, означающий продажу или возврат
Function SendReceipt(IReturn:boolean):integer;
var
  sName, sUnit:string;
  nQuant, nPrice, nCash,nCred1,nCred2,nCred3,nCred4:double;
  nVat, nTotDisc, nDisc:integer;
  PrintRecltem_proc: function(sN:string;nQuant,nPrice:double;
    nVat:integer;sUnit:string):integer;stdcall;
  BeginFiscalReceipt_proc: function():integer;stdcall;
  PrintCommentLine_proc: function(_Line : string; _attrib : integer):integer;stdcall;
  DiscountAdditionForItem_proc: function(_type : integer; _amount : double):integer;stdcall;
  DiscountAdditionForReceipt_proc: function(nType:integer;amount:double):integer;stdcall;
  EndReceipt_proc: function(_Pay,_K1,_K2,_K3,_K4:double):integer;stdcall;
begin
  //назначаем переменные для чека
  sName:='Item1';
  sUnit:='Kg.';
  nQuant:=3;
  nPrice:=1.18;
  nDisc:=10;
  nTotDisc:=5;
  nVat:=0;//0..3
  nCash:=3.03;
  nCred1:=0;
  nCred2:=0;
  nCred3:=0;
  nCred4:=0;
  //Получаем адрес процедуры начала чека
  @BeginFiscalReceipt_proc := GetProcAddress(FisHandle, 'BeginFiscalReceipt');
  result:=BeginFiscalReceipt_proc();//и вызываем
  if Result<>0 then
    exit;
  //комментарий
  @PrintCommentLine_proc := GetProcAddress(FisHandle, 'PrintCommentLine');
  result:=PrintCommentLine_proc('This is fiscal receipt!',65);//65-маска форматирования текста
  if Result<>0 then
```

```

    exit;
    //Товар в чеке.
    @PrintRecItem_proc := GetProcAddress(FisHandle, 'PrintRecItem');
    result:=PrintRecItem_proc(sName,abs(nQuant),abs(nPrice),nVat,sUnit);
    if Result<>0 then
        exit;
    {PrintRecItem в паре с DiscountAdditionForItem можно повторять друг за другом для каждого товара в чеке.
    DiscountAdditionForItem использовать необязательно
    }
    //Скидка на позицию
    @DiscountAdditionForItem_proc := GetProcAddress(FisHandle, 'DiscountAdditionForItem');
    //Первый параметр - тип скидки. 1 - это проценты
    //Отрицательное число - скидка. Положительное - наценка
    result:=DiscountAdditionForItem_proc(1,-1*nDisc);
    if Result<>0 then
        exit;
    //Скидка на чек. Использовать необязательно
    @DiscountAdditionForReceipt_proc :=
        GetProcAddress(FisHandle, 'DiscountAdditionForReceipt');
    //Первый параметр - тип скидки. 1 - это проценты
    result:=DiscountAdditionForReceipt_proc(1,-1*nTotDisc);
    if Result<>0 then
        exit;
    //В зависимости от параметра (флага) мы заканчиваем чек командой продажи или возврата
    if IReturn then
    //При возврате сумма денег должна совпадать с суммой чека и ВСЕ позиции
    //товаров должны иметь отрицательное количество
        @EndReceipt_proc := GetProcAddress(FisHandle, 'GoodsReturn')
    else
        @EndReceipt_proc := GetProcAddress(FisHandle, 'EndFiscalReceipt');
    result:=EndReceipt_proc(nCash,nCred1,nCred2,nCred3,nCred4);
end;

```

**ПЕЧАТЬ НЕ ФИСКАЛЬНОГО ЧЕКА:****Function SendNonFisReceipt():integer;**

```

var
    NonFiscalReceipt_proc: function():integer;stdcall;
    PrintNonFiscalLine_proc: function(_Line : string; _attrib : integer):integer;stdcall;
begin
    //Получаем адрес процедуры начала не фискального чека
    @NonFiscalReceipt_proc := GetProcAddress(FisHandle, 'BeginNonFiscalReceipt');
    result:=NonFiscalReceipt_proc();
    if Result<>0 then
        exit;
    //Печатаем 2 строки
    @PrintNonFiscalLine_proc := GetProcAddress(FisHandle, 'PrintNonFiscalLine');
    result:=PrintNonFiscalLine_proc('Line1',65);//65-маска форматирования текста
    if Result<>0 then
        exit;
    result:=PrintNonFiscalLine_proc('Line2',65);//65-маска форматирования текста
    if Result<>0 then
        exit;
    //Заканчиваем не фискальный чек
    @NonFiscalReceipt_proc := GetProcAddress(FisHandle, 'EndNonFiscalReceipt');
    Result:=NonFiscalReceipt_proc();
    if Result<>0 then
        exit;
end;

```

**ВЫГРУЗКА БИБЛИОТЕКИ:**

```
FreeLibrary(FisHandle);
```

**ВНИМАНИЕ!!!** Выгрузка библиотеки обязательна, т.к. она необходима для корректной работы фискала.

## 5. ЕС ДЕКЛАРАЦИЯ СООТВЕТСТВИЯ



### EC Declaration of Conformity

We, BRIO SRC, SIA, declare that product:

#### **FISKĀLAIS MODULIS**

Model: **BRIO-FisFM32/32E/M/U**  
is in conformity with 2006/95/EC (LVD Directive)

For the evaluation of the compliance with this Directive,  
the following standards or standardized documents  
were applied:

EN 60950-1:2006+ A11:2009 + A1:2010 + A12:2011  
- Information technology equipments  
- Safety

Person responsible for making this declaration

Name, Surname: Oleg Khalatov

Position/ Title: Director



(Signature)



LATVIA, RIGA

Oct/22/2012