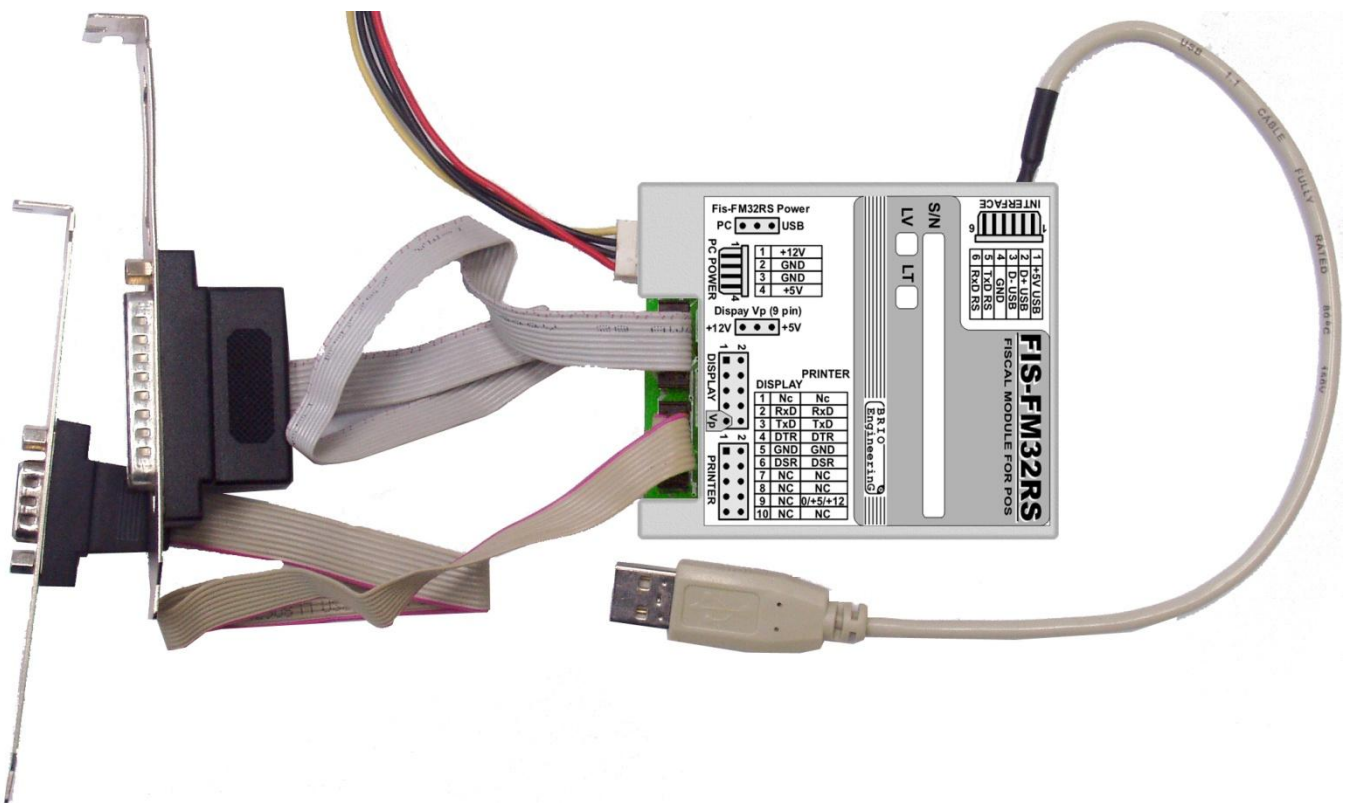


# BRIO Fis-FM32

## ФИСКАЛЬНЫЙ МОДУЛЬ ПРОГРАММИРОВАНИЕ МОДУЛЯ



О. Халатов

Фискальный модуль Fis-FM32™ V1.0 для POS систем  
Программирование модуля. Rev. 1  
BRIO EngineerinG, 2007,  
Рига, Латвия.



Данный документ содержит описание, порядок работы и технические характеристики фискального модуля Fis-FM32™. Модуль предназначен для работы в составе торговых POS систем, базирующихся на PC совместимых компьютерах, и полностью соответствует требованиям законодательства **Латвийской Республики**.

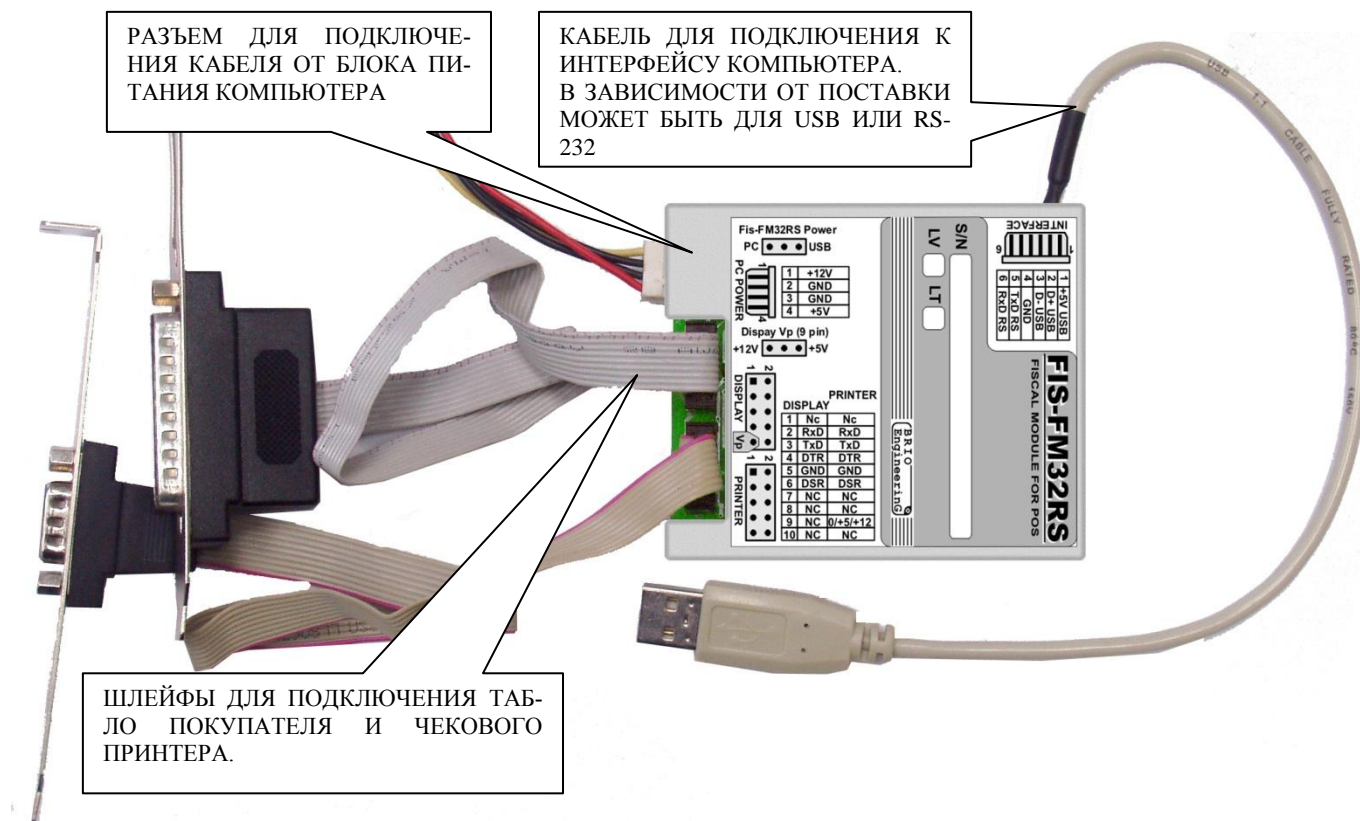
## ОГЛАВЛЕНИЕ

<b>1. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ.....</b>	<b>3</b>
<b>2. КАБЕЛИ И РАЗЪЕМЫ.....</b>	<b>4</b>
<b>3. СОСТОЯНИЯ ФИСКАЛЬНОГО МОДУЛЯ.....</b>	<b>5</b>
<b>4. БИБЛИОТЕКА ФУНКЦИЙ ДЛЯ РАБОТЫ С BRIO- FM32. ....</b>	<b>6</b>
<b>5. КОДЫ ОШИБОК .....</b>	<b>11</b>
<b>6. ПРОГРАММИРОВАНИЕ ФИСКАЛЬНОГО МОДУЛЯ .....</b>	<b>13</b>
6.1. НАЧАЛЬНАЯ НАСТРОЙКА FM32 .....	Ошибка! Закладка не определена.
6.2. ДОСТУПНОСТЬ ФУНКЦИЙ .....	13
6.3. ПОСЛЕДОВАТЕЛЬНОСТЬ ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ ФУНКЦИЙ .....	13
6.4. ПРИМЕР ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ ФУНКЦИЙ .....	13
<b>7. ЕС ДЕКЛАРАЦИЯ СООТВЕТСТВИЯ.....</b>	<b>16</b>

## 1. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

<b>Интерфейс связи с PC, с автоматическим выбором типа интерфейса.</b>	- USB V1.1, - USB V2.0, - RS-232C (115200 Bits/sec. 8 Bits, Non Parity, 1-Stop, Non flow control)
<b>Тип встроенного процессора</b>	- ARM. Программа закрыта "битом секретности"
<b>Максимальный ток потребления</b>	- 100 mA (Max)
<b>Объем фискальной памяти ROM</b>	- 8 MByte
<b>Объем памяти для электронной контрольной ленты</b>	- 4 MByte
<b>Режим обмена PC ↔ Fis-FM32</b>	- Пакетный обмен по специальному протоколу.
<b>Фискализация</b>	- Однократная.
<b>Максимальное количество Z-отчетов</b>	- 8192
<b>Инициализация</b>	- Однократная.
<b>Аппаратная диагностика</b>	- Встроенная система диагностики ошибок
<b>Защита от стирания</b>	- Специальная схема блокировки команды стирания.
<b>Дополнительные интерфейсы</b>	- RS-232C для подключения чекового принтера - RS-232C для подключения табло покупателя, с дополнительным выводом питания (5V или 12V) на 9-й вывод разъема.
<b>Типы поддерживаемых принтеров</b>	Epson-210 Epson-260 Epson-950 CHD- TH582 Samsung- SRP350 Samsung- SRP275 Citizen
<b>Типы поддерживаемых табло покупателя</b>	Любые, использующие интерфейс RS-232C

## 2. КАБЕЛИ И РАЗЪЕМЫ



### RS-232C/USB СВЯЗЬ С КОМПЬЮТЕРОМ

Конт.	Сигнал	Направление	Назначение
1	DC +5V USB		Напряжения питания +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

**ВНИМАНИЕ!!!** Фискальный модуль может питаться как от блока питания компьютера, так и от напряжения питания, поступающего по кабелю USB. Выбор осуществляется перемычкой Fis-FM32 POWER PC/USB. Но в случае питания от шины USB напряжения питания на 9 вывод интерфейса ТАБЛО ПОКУПАТЕЛЯ не подается.

### RS-232C ПРИНТЕР И ТАБЛО ПОКУПАТЕЛЯ

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

### 3. 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.

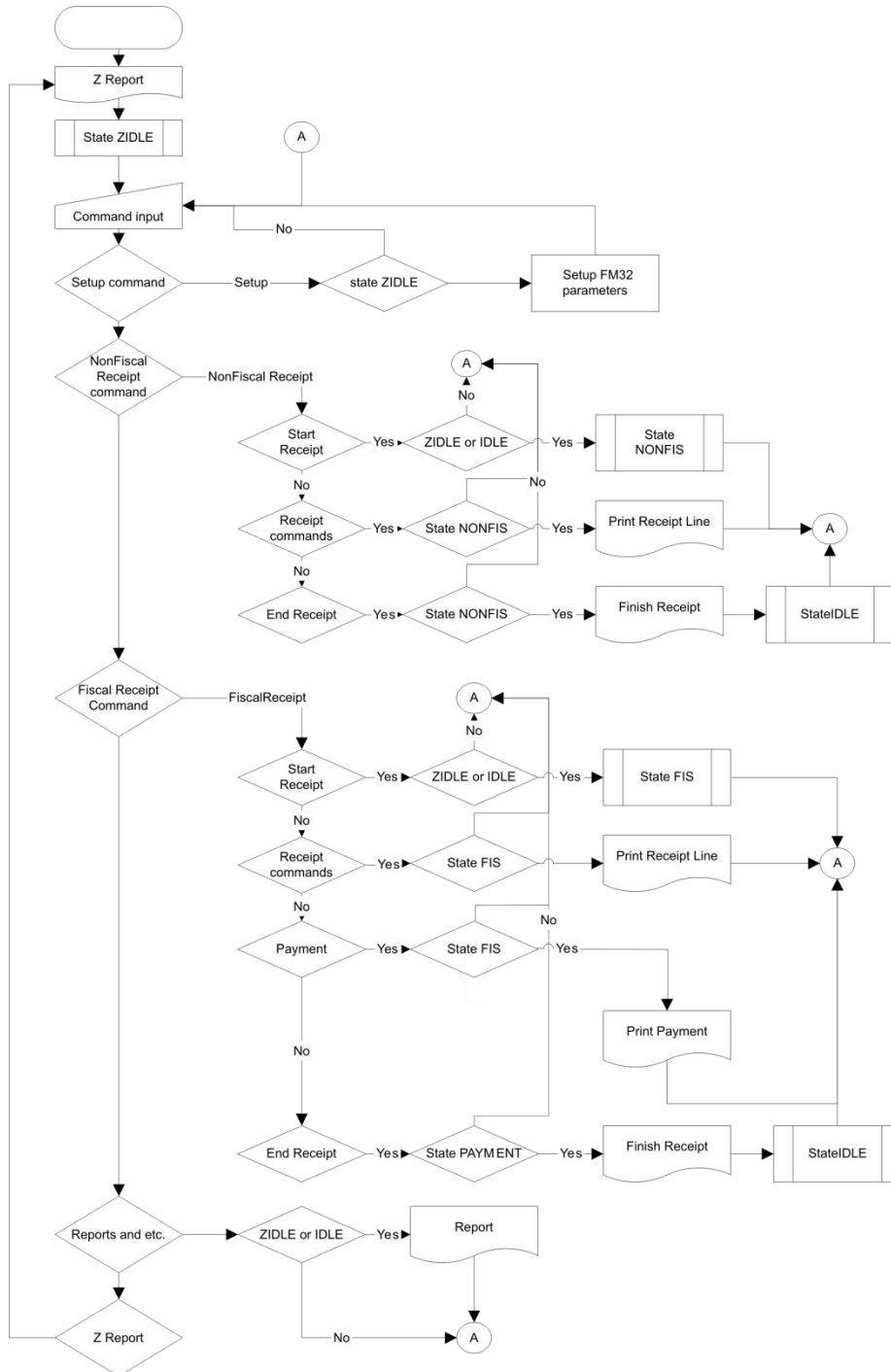
**ZIDLE** - Фискальный модуль переходит в это состояние, после выполнения Z-отчета.

**FIS** - Состояние фискального модуля, когда начато оформление нового фискального чека, но еще не оформлена его оплата.

**NONFIS** - Состояние, когда начат, но еще не завершен не фискальный чек.

**HARD** - Фискальный модуль переходит в это состояние в случае ошибки при работе с дисплеем или чековым принтером.

#### АЛГОРИТМ ПРОХОЖДЕНИЯ КОМАНД



## 4. БИБЛИОТЕКА ФУНКЦИЙ ДЛЯ РАБОТЫ С BRIO- FM32.

### 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 PrintRecItem( 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 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 PrintRecItemDepart( char * name, double quantity, double price,
```

```
int taxNumber, char * unit, int depart )
```

**ЗАПИСЬ НОМЕРА ШАССИ**

```
int SetShassi( char * number )
```

## 5. КОДЫ ОШИБОК

Fis-FM32 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_QUANTITY	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_DA	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_OVERFLOW	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_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	0x64	Несуществующий номер отдела
ERR_DEPART_DESC	129	0x64	Неправильное название отдела
ERR_DEPART_SET	130	0x64	Ошибка предустановки параметров отдела

<b>FiscalUnit.DLL ERRORS</b>	<b>ЗНАЧЕНИЕ</b>	<b>ПРИМЕЧАНИЕ</b>
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

## 6. ПРОГРАММИРОВАНИЕ ФИСКАЛЬНОГО МОДУЛЯ

### 6.1. 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().

### 6.2. 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

### 6.3. ПРИМЕР ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ ФУНКЦИЙ

Пример написан на 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;
  PrintRecItem_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 lReturn 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);**

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

## 7. ЕС ДЕКЛАРАЦИЯ СООТВЕТСТВИЯ

### ЕС Atbilstības deklarācija

Mēs, SIA "BRIO ZRF" ar pilnu atbildību, deklarējam, kas tālāk nosauktais produkts atbilst prasībām, kas noteiktas

**Zemsprieguma direktīvā  
73/23/EEC ar papildinājumiem 93/68/EEC**

Produkta kategorija: Fiskālais modulis

Modeļa nosaukums: BRIO Fis-FM32

Piemērotie standarti: EN60950, EN55022, EN55010

CE zīmes piestiprināšanas gads: 2006 g.

Rīga, 12.10.2006



**Oļegs Halatovs, Direktors**