

ЕСР Библиотека

ПРОГРАММНЫЙ ИНТЕРФЕЙС ДЛЯ ИНТЕГРАЦИИ С КАССОВЫМИ
РЕШЕНИЯМИ

1.0.6

Оглавление

1. Введение.....	3
2. Состав библиотеки.....	3
3. Настройка платежного приложения терминала	3
4. Настройка библиотеки	5
5. Программный интерфейс.....	7
6. Процесс выполнения операции	9
6.1. Взаимодействие с пользователем	9
6.2. Коммуникационные задачи.....	9
6.2.1. TLS.....	10
6.3. Печать чеков	10
6.4. Вывод результатов операции в файл.....	10
7. Незавершенные операции.....	11
8. Утилита «cashier_test».....	13
Приложение №1 Таблица кодов операций	16
Приложение №2 Коды результатов выполнения операций	18
Приложение №3 Параметры операций	20
Приложение №4 Пример файла cashier.ini	23
Приложение №5 Пример вызова библиотеки.....	24

Список изменений

Версия	Дата	Автор	Детали
1.0.1	31.03.2020	Исламов Р.Р.	Первая версия руководства.
1.0.2	21.04.2021	Лукьянов Д.А.	Обновлено приложение 1, 2, 3
1.0.3	17.03.2023	Сарматов А.А.	Добавлено Приложение 4. Изменен раздел 4 , Дополнен раздел 8
1.0.4	29.03.2023	Сарматов А.А.	Дополнен раздел №3 Параметры операций
1.0.5	07.04.2023	Сарматов А.А.	Добавлены команды в 1103 , 1104 в Приложение №1
1.0.6	17.04.2023	Сарматов А.А.	Изменены описания параметров вызова в Разделе 8. Добавлено описание выходного файла в разделе 6.

1. Введение.

Данный документ описывает библиотеку, разработанную для упрощения взаимодействия программного обеспечения кассы с терминалами, работающими под управлением платежного приложения «INVERSUM POS». Библиотека позволяет решить следующие задачи:

- Избавить разработчиков кассовых приложений от необходимости самостоятельно реализовывать командный протокол взаимодействия с терминалом.
- Обеспечить обратную совместимость, и минимизировать необходимость модификации кассового программного обеспечения в процессе развития протокола взаимодействия с терминалом.
- Предоставить разработчикам программного обеспечения кассы программный интерфейс для взаимодействия с терминалом.

2. Состав библиотеки

Библиотека состоит из следующих компонент:

- Конфигурационный файл **«cashier.ini»**
Содержит набор конфигурационных параметров, необходимых для работы библиотеки.
- Библиотека **«libecr_cashier.so»** для системы Linux или **«libecr_cashier.dll»** (x32 или x64) для системы Windows. (В дальнейшем будем называть ее просто **«libecr_cashier»**)
Библиотека, реализующая протокол взаимодействия с терминалом и предоставляющая программный интерфейс разработчикам кассового программного обеспечения.
- Исполняемый файл **«cashier_test»**
Утилита, позволяющая проверить работу кассы и терминала с текущими параметрами.

3. Настройка платежного приложения терминала

Для настройки режима работы с кассой в платежном приложении терминала необходимо выполнить операцию «Настройка кассы». Доступно три режима работы платежного приложения терминала:

- **Без кассы**
Терминал не принимает команды на выполнение операций от кассы. Запуск операции и ввод всех параметров необходимые для ее выполнения осуществляется с использованием пользовательского интерфейса платежного приложения терминала. Коммуникационные задачи, и задачи печати чеков выполняются с использованием оборудования самого терминала.

- **С кассой**
В данном режиме терминал функционирует так же, как и в режиме «Без кассы». Дополнительно терминал способен принимать и обрабатывать команды на выполнение операций от кассы. Если операция была инициирована кассой, печать чеков и коммуникационные задачи в зависимости от настройки могут выполняться средствами оборудования самого терминала, либо кассы.

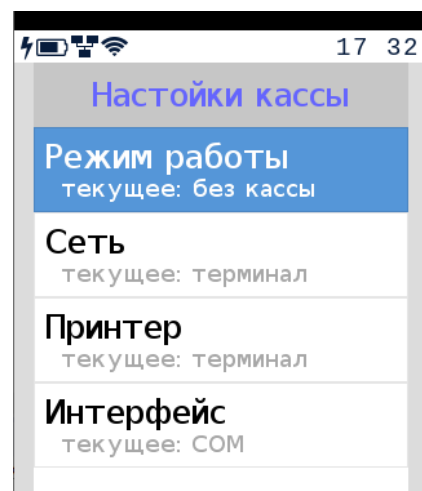


Рисунок 1. Настройка кассы

- **Только касса**

В данном режиме основная задача терминал - обслуживать команды, полученные от кассы. Запуск финансовых и технических (операций кассира) операций с использованием пользовательского интерфейса терминала невозможен. Но сохраняется возможность получить доступ к меню сервисных операций непосредственно через пользовательский интерфейс платежного приложения терминала. для корректировки конфигурации терминала, и корректировке параметров самого режима работы с кассой.



Поведение терминала в режиме работы «Только касса» может быть откорректировано в соответствии с пожеланиями заказчика. Возможно отключить доступ к меню технических операций, изменить набор операций доступных в меню, изменить структуру самого меню.

Меню настройки режима работы с кассой:

Пункт меню	Возможные значения	Описание
Режим работы	«Без кассы» «С кассой» «Только касса»	Описание режимов работы приведено выше
Сеть	«На терминале» «На кассе»	Задаёт, сетевой интерфейс какого устройства будет использовано в процессе выполнения операции: «На терминале» – будут использоваться коммуникационные интерфейсы доступные на терминале. «На кассе» – будут использоваться коммуникационные интерфейсы доступные на кассе.
Принтер	«На терминале» «На кассе»	Задаёт, принтер какого устройства будет использовано в процессе выполнения операции: «На терминале» – будут использоваться принтер терминала «На кассе» – будет использоваться принтер кассы
Интерфейс	«COM» «USB» «Ethernet»	Определяет интерфейс с помощью, которого осуществляется взаимодействие с кассой, и его конфигурационные параметры.



Терминал не взаимодействует с принтером и сетевым интерфейсом кассы непосредственно. Для отправки данных чека и обмена данными с удаленными серверами терминал использует команды, описанные протоколом взаимодействия терминала и кассы.



Список интерфейсов, доступных для соединения кассы и терминала может отличаться для разных моделей терминалов.

4. Настройка библиотеки

Конфигурационные параметры библиотеки «**libecr_cashier**» содержатся в файле «**cashier.ini**». Конфигурация определяет следующие параметры:

- Интерфейс подключения терминала к кассе (секция «Main»).
Доступно подключение терминала по USB либо RS232 или по Ethernet(LAN).
- Параметры печати чеков (секция «Receipt»).
Настраивается кодировка, которая будет использоваться библиотекой при обработке данных чека.
- Параметры вывода лога работы библиотеки (секция «Log»).
Библиотека может вести лог своей работы: информация о запущенной операции, результат выполнения операции, ошибки возникшие в процессе работы, обмен командами с терминалом. Данная группа параметров позволяет настроить уровень логирования, и метод записи лога.

Список параметров:

Параметр	Секция	Описание
Dialog	Main	Режим работы с диалогами: 0 – диалоги на терминале 1 – диалоги на кассе (используется внешняя библиотека)
Connection	Main	Строка, описывающая параметры подключения кассы к терминалу. Строка имеет следующий формат: com:<name>:<timeout>[:<baud_rate>:<parity>:<data_bits>:<stop bits>] name – имя COM-порта timeout – таймаут для операций записи и чтения в миллисекундах. baud_rate, parity, data_bits, stop bits – описывают параметры COM-порта. Данные параметры являются опциональными. Пример: COM порт на Linux: com:/dev/ttyACM0:5000
TimeSync	Main	Включить синхронизацию времени с кассой 0 – выключено 1 – включено
TimeZone	Main	Строка с временной зоной кассы Может содержать значения от -12 до 12
TcpID	Main	Задаёт уникальный идентификатор для связи POS и Кассы, в режиме работы TCP через «INVERSUM CREG»
Level	Log	Уровень логирования. Может принимать значение от 0 до 6, где 0 – минимальный а 6 – максимальный.
Type	Log	Способ записи лога: 0 – лог не используется 1 – лог выводиться в устройство стандартного вывода (stdout) 2 – запись лога в файл

		При выборе режима записи в файл, библиотека будет создавать файл с именем, равным текущему времени в папке «logs».
MaxCount	Log	Максимальное количество файлов для логов
Encoding	Receipt	Задаёт кодировку, в которой будут записываться данные чека. Поддерживаются два варианта: - cp1251 - utf8
MaxWidth	Receipt	Задаёт максимальную ширину чека
File	Receipt	Задаёт имя файла, куда будет сохранен чек в режиме работы «чек на кассе»
Transport	Key	3DES ключ шифрования трафика POS<->CASHIER

Пример файла настройки (cashier.ini) - в Приложении №4

5. Программный интерфейс

Программный интерфейс библиотеки включает в себя следующие методы:

- extern "C" ECR_CASHIER_API void* CashierCreate().

Описание	Создает новый объект операции, и возвращает указатель на него.
Параметры [IN]	N/A
Параметры [OUT]	N/A
Возвращаемое значение	- Указатель на созданный объект операции. - NULL в случае ошибки.

- extern "C" ECR_CASHIER_API void CashierDelete(void *obj).

Описание	Удаление объекта операции, созданного функцией «CashierCreate()».
Параметры [IN]	- obj - Указатель на объект операции.
Параметры [OUT]	N/A
Возвращаемое значение	N/A

- extern "C" ECR_CASHIER_API void CashierClear(void *obj).

Описание	Отчистка объекта операции, созданного функцией «CashierCreate()» от данных по выполненной операции. После вызова этой функции объект может быть использован повторно.
Параметры [IN]	- obj - Указатель на объект операции.
Параметры [OUT]	N/A
Возвращаемое значение	N/A

- extern "C" ECR_CASHIER_API int CashierSet(void *obj, const char *key, const char *value, int size).

Описание	Задаёт параметр операции.
Параметры [IN]	- obj – Указатель на объект операции - key – Имя параметра. NULL-терминированная ASCII-строка. - value – Указатель на значение параметра. - size – Длина значения параметра.
Параметры [OUT]	N/A
Возвращаемое значение	- 0 – успешно. - Другое значение в случае ошибки.
Примечание	Список параметров и формат их значений приведен далее. (см. Приложение №3)

- extern "C" ECR_CASHIER_API int CashierRun(void *obj, int cmd).

Описание	Иницирует процедуру выполнения операции с заданными параметрами на терминале.
Параметры [IN]	- obj – Указатель на объект операции - cmd – Код выполняемой операции
Параметры [OUT]	N/A
Возвращаемое значение	- 0 – успешно завершение операции - Другое значение в случае ошибки.
	Список ошибок, возвращаемых функцией «CashierRun()» приведен в Приложении №1 (см. Приложение №2). Список кодов операций приведен в Приложении №2 (см. Приложение №1).

- extern "C" ECR_CASHIER_API int CashierGet(void *obj, const char *key, char *value, int size).

Описание	Позволяет прочитать значение одного из параметров результата выполнения операции.
Параметры [IN]	- obj – Указатель на объект операции - key – Имя параметра. NULL-терминированная ASCII-строка. - size – Размер буфера для записи значения
Параметры [OUT]	- value – Указатель на буфер, для записи значения
Возвращаемое значение	- длина данных записанных в буфер - -1 – параметр не найден
Примечание	Если длина значения запрашиваемого параметра превышает размер переданного буфера (size). Функция возвращает длину значения. Данные в буфер при этом не записываются. Список параметров и формат их значений приведен далее. (см. Приложение №3)

- extern "C" ECR_CASHIER_API int CashierStop(void *obj).

Описание	Прерывает процедуру выполнения операции на терминале.
Параметры [IN]	- obj – Указатель на объект операции
Параметры [OUT]	N/A
Возвращаемое значение	- 0 – успешно - Другое значение в случае ошибки
Примечание	

Порядок вызова функций:

1. Кассовое приложение вызывает функцию **«CashierCreate()»** и получает указатель на объект операции.
2. Кассовое приложение заполняет параметры операции используя функцию **«CashierSet()»**
3. Кассовое приложение инициирует выполнение операции используя функцию **«CashierRun()»**
4. Кассовое приложение обрабатывает результат выполнения операции, читая необходимые ей параметры используя функцию **«CashierGet()»**
5. После завершения работы с объектом операции кассовое приложение может:
 - удалить объект текущей операции используя функцию **«CashierDelete()»**
 - очистить объект операции используя функцию **«CashierClear()»** и использовать его повторно при выполнении следующих операций.

6. Процесс выполнения операции

Библиотека **«libecr_cashier»** взаимодействует с платежным приложением терминала используя протокол, описанный в документе [«Интеграция с кассовыми решениями. Описание протокола взаимодействия кассы с терминалом»](#).

После передачи терминалу команды на выполнение операции (вызов функции **«CashierRun()»**), управление процессом выполнения заданной операции переходит к терминалу.



Операция считается выполненной успешно только если функция **«CashierRun()»** вернула значение **0** в качестве результата своей работы. Решение о статусе операции не должно приниматься по содержимому чека и каким-либо другим параметрам.

6.1. Взаимодействие с пользователем

Для отображения статуса выполнения операции и взаимодействия с держателем карты, терминал использует собственный графический интерфейс.

В зависимости от конфигурации загруженной на терминал с сервера параметризации ТМС, держателю карты могут отображаться следующие диалоги:

- Меню выбора приложения чиповой карты
- Ввод ПИН
- Отображение результата чтения карты и инструкции о дальнейших действиях («Следуйте инструкциям на телефоне», «Приложите одну карту», «Попробуйте еще раз или используйте другую карту»)

6.2. Коммуникационные задачи.

В зависимости от настроек режима работы с кассой заданными в платежном приложении терминала, возможно два варианта организации сетевого взаимодействия терминалом с внешними системами:

- Терминал использует собственные сетевые интерфейсы.

При необходимости обмена данными с удаленным сервером, терминал использует одно из собственных сетевых устройств.

- Терминал использует сетевой интерфейс кассы.
В данном режиме терминал, для обмена данными с удаленным сервером, использует специальный набор команд, предусмотренный протоколом взаимодействия с кассой:
 - команда на соединение;
 - команда на передачу данных;
 - команда на чтение данных;
 - команда на закрытие соединения;

Библиотека **«libecr_cashier»** в свою очередь реализует обработку этих команд, используя коммуникационные возможности кассы.

6.2.1. TLS

Не зависимо от выбранного способа обработки коммуникационных задач, поддержка TLS протокола реализуется на терминале. Таким образом, сетевой интерфейс кассы используется только для проксирования уже зашифрованных терминалом данных.

6.3. Печать чеков

В зависимости от настроек режима работы с кассой заданными в платёжном приложении терминала, возможно два варианта печати чеков:

- Терминал печатает чеки используя собственный принтер.
- Терминал использует принтер кассы.
В данном режиме терминал передает данные чека кассе используя специальный набор команд, предусмотренный протоколом взаимодействия с кассой.
Библиотека **«libecr_cashier»** сохраняет данные чека в файл **«check.txt»**.
Формат чека и его содержимое настраиваются в системе параметризации ТМС.
Данные чека передаются кассе в формате UTF8. Касса перекодирует их при сохранении в файл используя кодировку, заданную в конфигурационном файле **«cashier.ini»**.
После завершения процедуры выполнения операции программное обеспечение кассы может распечатать данные файла **«check.txt»**.

6.4. Вывод результатов операции в файл

По результатам выполнения операции (финансовой, административной, служебной) библиотека **«libecr_cashier»** формирует выходной файл в текущей папке .

Все выходные параметры представлены в формате

[название параметра] = [значение]

Названия, назначение и формат параметров представлены в Приложение №3 Параметры операций

Пример файла (outdata.txt) ,

```
aid=0000000041010
amount=10000
application_label=MasterCard
auth_code=192864
```

command=1
currency=643
entry_mode=6
expire=190630
host_date=140325
host_response_code=001
host_time=142734
mid=
pan=*****7692
paysystem_type=3
receipt_number=19
response_code=0
result_amount=10000
rrn=426427874714
serial_number=000302162304068
terminal_id=VERTEKS
transaction_date=140325
transaction_id=1681810958
transaction_status=0
transaction_time=142734
tvr=8020008001

7. Незавершенные операции

При возникновении сбоев в процессе выполнения операции после отправки терминалом авторотационного сообщения в процессинговый центр, терминал запоминает выполняемую операцию как «незавершенную». Сбои могут быть следующего характера:

- Потеря питания на терминале.
- Потеря связи с кассой.
- Отсутствие ответа от процессингового центра либо его некорректный формат.
- Любые прочие ошибки, в результате которых терминал был не в состоянии корректно завершить текущую операцию, и доставить результат ее выполнения кассе.

При наличии в журнале терминала «незавершенной» операции терминал блокирует выполнение последующих финансовых операций, и операции сверки итогов. На попытки их выполнения терминал в качестве результата вызова функции **«CashierRun()»** будет возвращать следующее значение:

- -27 - Есть незавершенная операция. (см. Приложение №2)

При получении данного кода, кассе необходимо периодически выполнять попытки завершения последней операции. Для этого кассе необходимо выполнять операцию:

- 1105 - [COMPLETION_LAST] Завершить последнюю операцию. (см. Приложение №1)

В процессе ее выполнения терминал делает попытку аннулировать последнюю операцию в процессинговом центре. Данная операция считается выполненной успешно если в процессе ее выполнения функция **«CashierRun()»** вернула одно из следующих значений :

- 0 - успех
- -13 - операция аннулирована

- -32 - нет незавершенной операции

Если несколько попыток завершения операции провалилось, для разблокировки работы терминала, касса может удалить данную операцию. Для этого кассе необходимо отправить команду на выполнение следующей операции:

- 1106 - [DELETE_LAST] Удалить последнюю операцию. (см. Приложение №1)

При выполнении данной операции терминал удаляет незавершенную операцию из своего журнала и возвращается к нормальному режиму работы. Данная операция не будет учтена при сверке итогов с процессинговым центром.

8. Утилита «cashier_test».

Данная утилита позволяет запускать операции на терминале используя библиотеку «libecr_cashier» без участия программного обеспечения кассы. Она может быть полезна в качестве инструмента тестирования и для выполнения некоторых сервисных операций на терминале, работающем в режиме работы с кассой.

Использование:

```
cashier_test [-?] [-help] [-o] [-a] [-b] [-d] [-c] [-i] [-x] [-r] [-v]
              [-q] [-s] [-rd] [-dd] [-pd] [-bi] [-se] [-vi]
```

Description:

- ? - Show help information message
- help - Show help information message
- o - (command) Operation command
- a - (amount) Operation amount without any delimiters
- b - (amount_cashback) Cashback amount. Only for operation Purchase with Cashback
- d - (amount_cert) Operation amount by certificate. Only for operation Purchase with EC
- c - (currency) Operation currency code
- i - (terminal_id) Terminal identification
- x - (transaction_id) Transaction ID
- r - (rrn) Retrieval reference number
- v - (auth_code) Authorization code
- q - (receipt_number) Receipt number
- s - (account_type) Account type
- rd - (receipt_data) File name for print on POS
- dd - (dialog_data) Dialog properties string
- pd - (proxy_data) Proxy link data string
- bi - (basket_id) Basket ID. Only for operation Purchase with EC
- se - (stage_ec) Stage EC operation: 1 (verification), 2 (auth), 3 (pan status inquiry), 4 (refund auth), 5 (return)
- vi - (vcard_id) Virtual card id, configured on TMS -Virtual terminal

Example of usage:

- purchase operation (command 1, amount 1.23, currency 643):

```
cashier_test -o=1 -a=123 -c=643
```

Параметры используемые при запуске утилиты «cashier_test»:

N – цифровое значение

AN – цифро-буквенное значение

Параметр	Значение	Описание
o	N	Код запускаемой операции. Список всех кодов операций приведен в разделе «Приложение 1» Пример: -o=1
a	N	Сумма операции в минимальных единицах валюты, без разделителя. Пример: -a=10000
c	N	Трёхзначный международный код валюты.

		Пример: -с=643
i	AN	Идентификатор терминала. Данный параметр является опциональным. Его необходимо задавать при выполнении технических операций («Сверка итогов», «Тест связи», и т. д.) если терминал поддерживает работу с несколькими процессинговыми центрами. Пример: -i=TPTPRET
r	AN	RRN оригинальной операции (Retrieval Reference Number). Данный параметр может передаваться при выполнении операций «Возврат», «Коррекция» для задания связи с оригинальной операцией. Пример: -r=123456789
v	AN	Код авторизации оригинальной операции. Данный параметр может передаваться при выполнении операций «Возврат», «Коррекция», «Завершение расчета» для задания связи с оригинальной операцией. Пример: -v=123456789
q	N	Номер чека оригинальной операции. Данный параметр необходимо передавать при выполнении операции отмены. Пример: -q=999
s	N	Тип счета. Данный параметр является опциональным. Определяет по какому счету будет выполнена финансовая операция. Может принимать одно из следующих значений: 1 – по умолчанию 2 – текущий (чековый) 3 – сберегательный 4 – кредитный Пример: -s=1
x	N	Идентификатор оригинальной операции (Transaction ID). Данный параметр может передаваться при выполнении операций «Возврат», «Отмена», «Коррекция» для задания связи с оригинальной операцией. Пример: -x=123456789012345678
b	N	Сумма выдачи наличных на кассе ТСП в минимальных единицах валюты, без разделителя. Пример: --amount_cashback10000
d	N	Сумма Электронного сертификата в минимальных единицах валюты, без разделителя. Пример: -ad=10000

bi	N	Идентификатор корзины товаров для оплаты средствами Электронного сертификата и собственными средствами Пример: -bi=123456789012345678
rd	AN	Имя текстового файла на кассовом ПК для печати на принтере терминала Пример: -rd=File.txt
sd	AN	Вывод строкового диалога на экран терминала
vi	N	Идентификатор виртуальной карты (в памяти терминала) для выполнения операции Пример: -vi=1

Примеры операций:

Тест связи с процессинговым центром: cashier_test -o=1113

Загрузка конфигурации с ТМС: cashier_test -o=1118

Оплата 50.00 Руб. : cashier_test -o=1 -a=5000 -c=643

Возврат 100.00 Руб. : cashier_test -o=2 -a=10000 -c=643 -r=123456789012

Приложение №1 Таблица кодов операций

Таблица кодов операций.

Код	Название	Описание
1	PURCHASE	Оплата
2	REFUND	Возврат
3	PREAUTH	Предавторизация
4	PREAUTH_COMPLETION	Завершение предавторизации
5	CASH_ADVANCE	Выдача наличных
6	CASH_DEPOSIT	Внесение наличных
7	BALANCE	Баланс
8	PIN_CHANGE	Смена PIN
9	CANCEL	Отмена
10	QUASICASH	QUASICASH
11	PURCHASE_WITH_CASHBACK	Оплата с выдачей наличных
12	ADJUSTMENT	Коррекция
13	MINI_STATEMENT	Мини выписка
14	PAYMENTS	Платежи
15	CARD_VERIFICATION	Проверка карты
16	MAIL_ORDER	Оплата без держателя карты
21	PREAUTH_INCREMENT	Инкрементальная предавторизация
23	SMART_CANCEL	Умная отмена
24	PURCHASE_EC	Оплата Электронным сертификатом
26	PURCHASE_SBP	Оплата СБП
27	REFUND_SBP	Возврат СБП
28	CARD_READ	Чтение карты
1001	CANCEL_WITH_CARD	Отмена с предъявлением карты
1002	CANCEL_WITHOUT_CARD	Отмена без предъявления карты
1003	BALANCE_ON_RECEIPT	Баланс отображается в чеке
1004	BALANCE_ON_DISPLAY	Баланс отображается на экране терминала
1101	MAINCASHIER_MENU	Меню старшего кассира
1102	ADMIN_MENU	Меню администратора
1103	CASHIER_MENU	Меню кассира
1104	REVERT_LAST	Отмена последней успешной операции (Техническая)
1105	COMPLETION_LAST	Завершить последнюю операцию
1106	DELETE_LAST	Удалить последнюю операцию
1107	FULL_REPORT	Полный отчёт
1108	SHORT_REPORT	Краткий отчёт
1109	RECEIPT_DUPLICATE	Печать дубликата последнего чека
1110	RECEIPT_BY_ID	Печать дубликата чека по номеру
1111	RECONCILIATION	Сверка итогов
1112	COMBINED	Сверка итогов и загрузка параметров с ТМС и хоста
1113	CONNECTION_TEST	Тест связи
1117	DOWNLOAD_PARAMETERS	Загрузка параметров с хоста
1118	TMS_UPDATE	Загрузка параметров с ТМС, с текущими параметрами
1120	CASHIER_LOGIN	Авторизация кассира
1123	CLOSEDAY_TMSUPDATE	Сверка итогов и загрузка параметров с ТМС
1124	CHECK_QR_STATUS	Проверка статуса QR кода
1200	TERMINAL_STATE	Проверка соединения с терминалом
1202	PRINT_REQUEST	Печать файла с кассы на принтере терминала

1203	PROXY_LINK_REQUEST	Команда за использование коммуникаций терминала с кассы
1204	IDLE_ACTION	Команда для выполнения периодических событий на терминале (выгрузка данных в мониторинг, расписаний ТМС и сверки итогов)

Приложение №2 Коды результатов выполнения операций

Таблица результатов выполнения операции:

Код	Описание
0	Успешное завершение операции
-1	Системная ошибка
-2	Операция прервана
-3	Операция прервана
-4	Ошибка записи операции в журнал
-5	Журнал операций заполнен – необходимо выполнить сверку итогов
-6	Текущий период (смена) уже закрыта
-7	Необходимо выполнить сверку итогов
-8	Операция уже отменена
-9	Операция не может быть отменена
-10	Таймаут ожидания карты
-11	Операция не поддерживается
-12	Ошибка авторизации
-13	Операция была аннулирована
-14	Касса не доступна
-16	Карта не поддерживается
-17	Операция не может быть откорректирована
-18	Ошибка конфигурации терминала
-19	Некорректный формат данных карты
-20	Ошибка верификации держателя карты
-21	Ошибка открытия периода (смены)
-22	Слишком большая сумма
-23	Ошибка ввода ПИН
-24	Необходима загрузка ключей
-25	Карта просрочена
-26	Некорректный номер карты
-27	Есть незавершенная операция
-28	Отклонено процессинговым центром
-29	Операция не поддерживается
-30	Валюта не поддерживается
-31	На терминал не загружена конфигурация
-32	Операция не найдена
-33	Невозможно выполнить возврат
-34	Операция уже отменена
-35	Нет обновлений
-36	Не забрали карту
-37	Ошибка печати
-38	Неверная подпись
-39	Операция запрещена
-40	Хост недоступен
-41	Недостаточно flash-памяти для получения обновления
-42	Не указан TID для данной операции
-43	Терминал ожидает зашифрованные данные от кассы, касса начала обмен без шифрования.
-44	В команде от кассы терминалу отсутствует код запрашиваемой операции
-45	Команда от кассы содержит неизвестный терминалу код операции
-46	Отсутствует лицензия

-47	Недостаточно флэш памяти
-48	Отсутствует мастер ключ
-49	Операция в обработке
-50	Выполняется опрос статуса СБП
-51	Ошибка при шифровании данных с кассой. Возможно, ключ на кассе и терминале не совпадают

Приложение №3 Параметры операций

Таблица параметров операции, которые могут быть заданы функцией «CashierSet()».

Значение всех параметров задается в виде ASCII строки:

N – цифровое значение

AN – цифро-буквенное значение

Имя параметра	Формат значения	Описание
amount	N Длина: 1 – 12	Сумма операции. Задается в минимальных единицах валюты.
currency	N Длина: 3	Международный трехзначный цифровой код валюты.
amount_cert	N Длина: 1 – 12	Сумма операции. Задается в минимальных единицах валюты.
amount_cashback	N Длина: 1-12	Сумма выдачи наличных на кассе ТСП . Задается в минимальных единицах валюты.
rrn	AN Длина: 1 – 12	RRN оригинальной операции (Retrieval Reference Number). Данный параметр может передаваться при выполнении операций «Возврат», «Коррекция» для задания связи с оригинальной операцией.
transaction_id	AN Длина: 1-18	Уникальный Идентификатор транзакции (для процессинга TPTP (Compass+))
auth_code	AN Длина: 6	Код авторизации оригинальной операции.
terminal_id	AN Длина: 1 – 12	Идентификатор терминала в процессинговом центре.
receipt_number	N Длина: 1 – 5	Номер чека оригинальной операции.
basket_id	AN Длина: 1 – 24	Идентификатор КОРЗИНЫ ТОВАРОВ для оплаты Электронным Сертификатом НСПК
stage_ec	N Длина: 1	Запуск одного этапа операции оплаты Электронным Сертификатом
account_type	N Длина: 1	Тип счета.
vcard_id	N Длина: 1	Индекс виртуальной карты в параметрах терминала, полученных с ТМС

Таблица параметров результата выполнения операции, которые могут быть прочитаны функцией «CashierGet()»:

Имя параметра	Формат значения	Описание
pan	AN Длина: 16 – 19	Маскированный номер прочитанной карты. Параметры маски задаются конфигурацией терминала, загруженной с ТМС
expire	N Длина: 6	Срок действия карты в формате ГГММДД
aid	AN Длина: 10 – 32	Идентификатор приложения карты (AID). Параметр заполняется только для EMV карт
application_label	AN Длина: 1 – 24	Название приложения карты. Параметр заполняется только для EMV карт
tvr	AN	TVR (EMV Terminal Verification Result).

	Длина: 10	Параметр заполняется только для EMV карт
host_response_code	N Длина: 1 – 3	Код ответа, полученный от процессингового центра. Параметр заполняется только для онлайн операций.
auth_code	AN Длина: 6	Код авторизации, полученный от процессингового центра. Параметр заполняется только для онлайн операций.
rrn	AN Длина: 1 – 12	RRN (Retrieval Reference Number), полученный от процессингового центра. Параметр заполняется только для онлайн операций.
terminal_id	AN Длина: 1 – 12	Идентификатор терминала в процессинговом центре. Если идентификатор терминала был задан в качестве одного из параметров выполнения операции, то значение в результатах операции будет идентичным.
currency	N Длина: 3	Код валюты выполненной операции. Если код валюты был задан в качестве одного из параметров выполнения операции, то значение в результатах операции будет идентичным.
receipt_number	N Длина: 1 – 5	Номер чека, присвоенный терминалом операции.
transaction_time	N Длина: 6	Время операции на терминале в формате ЧЧММСС
transaction_date	N Длина: 6	Дата операции на терминале в формате ГГММДД
host_time	N Длина: 6	Время операции на хосте в формате ЧЧММСС
host_date	N Длина: 6	Дата операции на хосте в формате ГГММДД
transaction_id	AN Длина: 1-18	Уникальный Идентификатор транзакции (для процессинга TPTP (Compass+))
original_amount	N Длина: 1 – 12	Исходная сумма операции (без учета скидок/комиссий). Задается в минимальных единицах валюты.
result_amount	N Длина: 1 – 12	Финальная сумма операции(с учетом офлайн скидки/комиссии). Задается в минимальных единицах валюты.
cardholder	ANS Длина: 27-45	Имя держателя карты.
hashed_pan	AN Длина: 64	SHA256 – хеш от pan
card	ANS Длина: var 40	Результат СПЕЦИАЛЬНОЙ ОПЕРАЦИИ чтения карты PAN / PAN and EXP DATE / TRACK2 (тип данных параметризуется в TMC)
card_par	ANS Длина: 29	Ссылка на платежный счет (PAR) , EMV tag 9F24
entry_mode	N Длина: 1	Индикатор способа чтения карты клиента
Mid	ANS Длина: 1-15	Идентификатор Торговой точки установки терминала
serial_number	ANS Длина: var 32	Серийный номер банковского терминала
Mcc	N Длина: 4	MCC коммерсанта

paysystem_type	N Длина: 1	Идентификатор платежной системы (МИР,VISA, ЕС/МС,AMEX....)
----------------	---------------	---

Приложение №4 Пример файла cashier.ini

Пример файла настройки (cashier.ini)

[Main]

0 - console, 1 - extern DLL

Dialog=0

Time synchronization with ECR

0 - turn off, 1 - turn off

#TimeSync=0

#TimeZone=3

#TcpID=

connection for POS

COM connection: com:<index>[:<timeout>[:<baud rate>[:<parity>[:<data bits>[:<stop bits>]]]]]

TCP connection: tcp:<ip>:<port>[:<timeout>]

#Connection=com:COM6:300

#Connection=com:COM1:300:115200

Connection=tcp:10.0.0.102:5559

[Log]

Level 0 – critical, 1 – error, 2 warning, 3 – information, 4 – debug, 5 - trace

Level=5

0 - no log, 1 - console, 2 - file "cashier.log"

Type=1

max file count in logs folder, delete older files by modification time

MaxCount=30

[Receipt]

#Encoding=cp1251

#MaxWidth=40

#File=check.txt

#OutDataFile=outdata.txt

[Key]

#Transport=0123456789ABCDEFFEDCBA9876543210

Приложение №5 Пример вызова библиотеки

```
#include <stdio.h>
#include <stdexcept>
#include <string>
#include <iostream>
#include <sstream>
#include <iomanip>
#include <vector>
#include <map>
#include <set>
#include <algorithm>
#include <pthread.h>

#include "platform.h"

#define GET_DLL_FUNC(x, y, z) x = (y) GetProcAddress((HMODULE) extern_dll, z); if (x ==
NULL) throw std::runtime_error(z);

typedef void*(*_CashierCreate)();
typedef void(*_CashierDelete)(void*);
typedef void(*_CashierClear)(void*);
typedef int(*_CashierSet)(void*, const char *, const char *, int);
typedef int(*_CashierGet)(void*, const char *, const char *, int);
typedef int(*_CashierRun)(void*, int);
typedef int(*_CashierStop)(void*);

static _CashierCreate fCashierCreate      = NULL;
static _CashierDelete fCashierDelete      = NULL;
static _CashierClear fCashierClear        = NULL;
static _CashierSet fCashierSet            = NULL;
static _CashierGet fCashierGet            = NULL;
static _CashierRun fCashierRun            = NULL;
static _CashierStop fCashierStop          = NULL;

static bool stop_wait_cancel = false;

void * start_wait_cancel(void *cashier)
{
    uint64_t dead_line = GetTickCount() + 10000;

    while (!stop_wait_cancel && (dead_line > GetTickCount())) {
        Sleep(10);
    }

    if (!stop_wait_cancel && (dead_line <= GetTickCount())) {
        printf("Send CANCEL\n");
        fCashierStop(cashier);
    }
}
```

```

    return 0;
}

const char dll_name_win[] = "libecr_cashier.dll";
const char dll_name_linux[] = "./libecr_cashier.so";

int main(int argc, char *argv[])
{
    setbuf(stdout, NULL);

    void *extern_dll = LoadLibrary(dll_name_win);

    if (extern_dll == NULL) {
        extern_dll = LoadLibrary(dll_name_linux);

        if (extern_dll == NULL) {
            printf("ERROR: DLL file not open: %s (%s)\n", dll_name_win, dll_name_linux);
            printf("ERROR: %s\n", GetLastError());
            return -1;
        }
    }

    try {
        GET_DLL_FUNC(fCashierCreate, _CashierCreate, "CashierCreate");
        GET_DLL_FUNC(fCashierDelete, _CashierDelete, "CashierDelete");
        GET_DLL_FUNC(fCashierClear, _CashierClear, "CashierClear");
        GET_DLL_FUNC(fCashierSet, _CashierSet, "CashierSet");
        GET_DLL_FUNC(fCashierGet, _CashierGet, "CashierGet");
        GET_DLL_FUNC(fCashierRun, _CashierRun, "CashierRun");
        GET_DLL_FUNC(fCashierStop, _CashierStop, "CashierStop");
    }
    catch (std::exception &e) {
        printf("ERROR: %s\n", e.what());
        return -1;
    }

    // Create cashier object
    void *cashier = fCashierCreate();
    char response_buffer[4 * 1024];

    if (cashier == NULL) {
        printf("ERROR: NO CASHIER !!!\n");
        return -1;
    }

    try {
        // Clear cashier object data before using
        fCashierClear(cashier);

        // Set parameters for operation
        fCashierSet(cashier, "amount", "10000", 5);
    }

```

```

fCashierSet(cashier, "currency", "643", 3);

stop_wait_cancel = false;
pthread_t thread_id = NULL;
pthread_attr_t _attr;

if (0) {
    pthread_attr_init(&_amp;_attr);
    pthread_create(&thread_id, &_amp;_attr, start_wait_cancel, cashier);
    pthread_detach(thread_id);
    printf("Wait 10 seconds for CANCEL\n");
}

// RUN operation Purchase (code 1)
int res = fCashierRun(cashier, 1);
stop_wait_cancel = true;

if (thread_id) {
    Sleep(20);
    pthread_attr_destroy(&_amp;_attr);
    printf("Wait cancel stopped\n");
}

if (res != 0) {
    printf("result: %d\n", res);
}

// Read output parameters
fCashierGet(cashier, "response_code", response_buffer, sizeof(response_buffer));

if (std::string(response_buffer) != "0") {
    printf("response code: %s\n", response_buffer);
}

// Delete cashier object
fCashierDelete(cashier);
}
catch (std::exception &e) {
    printf("ERROR: exception: %s\n", e.what());
    return -1;
}
catch (...) {
    printf("ERROR: unknown exception\n");
    return -1;
}

return 0;
}

```