

ARCUS 2.1

для интеграции с кассовыми решениями

Руководство администратора

/ Версия 1.0

Содержание

Хронология изменений	4
Условные обозначения	4
Глоссарий	4
Аббревиатуры	6
Введение	7
1_ ARCUS 2.1 CAP	8
1_1 Состав программы, основные принципы ее функционирования.....	8
1_1_1 Использование TLS шифрования в режиме IP через кассу	9
1_2 Режим работы с кассой	9
1_2_1 Файл настроек ARCUS.CFG для Режимы работы с кассой терминального приложения ISO (Openway, SmartVista, Tieto, CTL)	10
1_2_2 Файл настроек ARCUS.CFG для Режимы работы с кассой терминального приложения Unipay (CompassPlus, SPDH).....	12
1_3 Интеграция с ККМ	14
1_3_1 Формат и содержание файла данных output.dat	16
1_4 Вызов ARCUS 2.1	17
1_4_1 Windows / Linux	17
1_5 Настройки ARCUS 2.1 CAP	19
1_5_1 Файл инициализации Cashreg.ini	19
1_5_2 Файл ops.ini для ISO8583.....	19
1_5_3 Файл ops.ini для Compass+ и SPDH.....	21
1_5_4 Файл rc_conv.ini	22
1_5_5 Файл rc_res.ini	22
1_6 Эксплуатация	23
1_6_1 Интеграция ARCUS 2.1 CAP с ПО ККМ	23
1_6_2 Виртуальная клавиатура.....	23
1_6_3 Способы и периодичность контроля правильности работы ПО	24
1_6_4 Аварийные ситуации и способы их устранения.....	24
1_6_5 Описание фрагментов файла лога	24
1_6_6 Возврат без предъявления карты.....	24
2_ Интеграция через RS232 или USB	26
2_1 Схема взаимодействия с ПО ККМ	26
3_ Низкоуровневая интеграция	27
4_ Arcus NET Server	28
4_1 Arcus NET Server v.1.3	28
4_1_1 Файл Server.conf	29
4_1_2 Файл Pinpads.conf	29

4_1_3	Файл Cashregisters.conf.....	30
4_1_4	Файл Log.conf	30
4_1_5	Пример настройки ARCUS NET SERVER v.1.3.....	31
4_2	Arcus NET Server v.2	32
4_2_1	Файловый режим	32
4_2_2	Описание файлов	33
4_2_3	Режим TCP/IP.....	35
4_2_4	properties-file Файлы настройки	35
4_2_5	Пример настройки ARCUS NET Server v.2.0.....	38
4_2_6	Пример файла лога	41
4_2_7	Пример настройки SSL	41
4_2_1	Аварийная отмена покупки	42
5_	Основные этапы установки и настройки ARCUS 2.1 CAP и ARCUS Net Server v.1.3.....	43
5_1	Тестовая транзакция (Windows и Linux)	44
Приложение 1 Ключи для методов set/get.....		45
Приложение 2 Пример использования библиотеки.....		45
	Прототипы используемых функций.....	Ошибка! Закладка не определена.
	Пример использования на языке C++	46
	Пример использования на языке C#.....	48
	Пример использования на языке Object Pascal (Delphi):	49
Приложение 3 Минимальный перечень поддерживаемых операций		52
Приложение 4 Информация о документе.....		53
	Список таблиц.....	53
	Список рисунков	53

Хронология изменений



/ Таблица 1. Хронология изменений

Версия	Дата	Автор	Детали
1.00	19/05/2017	Исламов Р.	Создание документа на основе документации на ARUCS2

Условные обозначения

Следующие условные обозначения использовались в документе:

/ Таблица 2. Условные обозначения

Обозначение	Значение
	Обратите внимание!
	Обратите особое внимание!

Глоссарий

/ Таблица 3. Глоссарий

Термин	Определение
EMV	Стандарт взаимодействия чиповых карт для проведения платежей по карте (составлено из первых букв компаний Europay, MasterCard and VISA).
EMV Приложение	Микропрограмма и блок параметров на карточке, относящейся к международной платежной системе EMV.
ПИН-пад	Электронное устройство, предназначенное для ввода PIN кода.
PIN код	Секретный код, вводимый клиентом (держателем карточки), для подтверждения того, что именно он обладает полномочиями по использованию карточки.
POS-терминал, терминал	Аппаратно-программный комплекс, предназначенный для регистрации факта приобретения товара и упрощения процедуры приобретения товара.
Код действия терминалов (ТАС)	Действия ПО терминала при возникновении конкретной ситуации (отказ от операции, проведении операции онлайн или действие по умолчанию).
Коммуникационный скрипт	Минипрограмма (последовательность команд), определяющая сценарий подключения терминала к серверу авторизации или серверу TMS.
Клиент/ Держатель	Физическое лицо (в том числе уполномоченный представитель юридического лица), на имя которого в силу заключенного между ним и Эмитентом договора и/или на имя которого по указанию лица, заключившего договор с Эмитентом, выпущена Карта.
Магнитная полоса	Полоса, содержащая идентификационную информацию о карте и ее держателе.

Термин	Определение
Операция	Расчетная операция, осуществляемая с использованием Карты с обязательной Авторизацией в соответствии с законодательством Российской Федерации, нормативно-правовыми актами Банка России и договором с Эмитентом, проводимая в Торговой точке по требованию Держателя.
Операция возврата	Расчетная операция, совершаемая с использованием Карты при возникновении необходимости в возврате Держателю денежных средств, списанных с его счета в рамках ранее совершенной Операции.
Операция отмены	Процедура отмены Кода Авторизации до формирования Электронного журнала.
Отчет Электронного терминала (Электронный журнал)	Документ или совокупность документов в электронной форме, сформированные за определенный период времени при совершении Операций/Операций с использованием POS-терминала.
Программное обеспечение (ПО)	Программное обеспечение POS-терминала.
Профиль	Один из объектов системы, предназначенный для хранения настроек групп параметров, определяющих алгоритм работы терминала в процессе выполнения операции с картой или административной операцией.
Реквизиты Карты	Номер Карты, срок действия Карты, код безопасности CVV2/CVC2/Batch code/CVV II/CAV2 (для Карт Платежных систем Visa International/MasterCard Worldwide/American Express/Diners Club International/JCB International соответственно).
Сервисный код карты	Код карты, который указывается область ее действия и тип.
Сервисный код коммерсанта (МСС)	Четырехзначный номер, который обозначает профиль деятельности, которым занимается коммерсант.
Скрипт эмитента (сценарий)	Команда или командная строка, передаваемая от эмитента к терминалу с целью выполнения команд(ы) микропроцессорной картой.
Способ верификации владельца карты	Способ верификации того, что человек, предъявляющий карту, имеет права на ее использование (например, сверка подписи или введение PIN кода).
Справочник	Один из объектов системы, предназначенный для хранения основных значений, шаблонов, терминов и алгоритмов, которыми оперирует система.
Транзакция	Группа последовательных операций, которая представляет собой логическую единицу работы с данными. Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще и тогда она не должна произвести никакого эффекта. Транзакции обрабатываются транзакционными системами, в процессе работы которых создается история транзакций.
Хост, коммуникационный сервер	Сервер базы данных платежной системы, который отвечает на запросы POS-терминала точки обслуживания, а также снабжает его необходимой информацией.
Эквайер	Банк или компания, осуществляющая весь спектр операций по взаимодействию с точками обслуживания карточек, которая состоит из терминалов в торгово-сервисной сети и банкоматов.
Эмитент	Организация, выпустившая (эмитировавшая) карточки для развития и финансирования своей деятельности.

Аббревиатуры

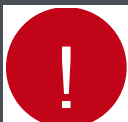
/ Таблица 4. Аббревиатуры

Аббревиатура	Расшифровка
AID	Application Identification Number (Идентификационный номер приложения).
AVN	Application Version Number (Номер версии приложения).
BIN	Bank Identification Number (Банковский идентификационный номер).
CAP	Cash register Application.
CVM	Cardholder Verification Method (Способ верификации владельца карты).
DDOL	Dynamic Data Object List (Список элементов данных необходимых для динамической аутентификации).
ICC	Integrated Circuit(s) Card (Карта с интегрированными электронными схемами).
ID	Identifier (идентификатор).
ISO	International Organization for Standardization (Международная организация стандартизации).
MCC	Merchant Category Code (Сервисный код коммерсанта).
MS	Magnetic Stripe (магнитная полоса).
TAC	Terminal Action Code(s) (Код(ы) действия терминалов).
PAN	Primary Account Number (Индивидуальный номер карты).
PIN	Personal Identification Number (Персональный идентификационный номер).
POS	Point of Sale (точка продажи, сервиса).
RID	Registered Application Provider Identifier (Зарегистрированный номер провайдера приложения; например, VISA, MasterCard).
TDOL	Transaction Data Object List (Список данных о транзакции участвующих при вычислении сертификата).
TMM	Terminal Management Mode (модуль управления терминалами).
TMS	Terminal Management System (система управления терминалами).
СУБД	Система управления базами данных.

Введение

Данное руководство, разработано компанией ООО «Инженико» и описывает этапы установки и настройки ARCUS2.1 – библиотеки, позволяющей проводить интеграцию Терминального платежного приложения Universal EMV POS¹ и кассового программного обеспечения. В документе описаны:

- ARCUS 2.1 CAP – совокупность программно-аппаратных средств, позволяющих обеспечить прием безналичных платежей на контрольно-кассовых машинах (ККМ) по банковским картам, включая магнитные, чиповые карты (EMV), бесконтактные карты, а также при ручном вводе данных;
- ARCUS NET SERVER – приложение, которое совместно с ARCUS 2 CAP, обеспечивает подключение терминала к ККМ по интерфейсу ETH, в том числе для ККМ, реализованных по схеме «тонкий клиент». Приложение реализовано в двух версиях.



Документ содержит описание программного продукта ARCUS 2.1 и является дополнением к документу Руководство администратора ARCUS 2.

Широкие возможности конфигурирования обеспечивают гибкость в настройке при наличии разнообразных требований к функциональности.

ПО соответствует основным требованиям и стандартам международных платежных систем.

Пользователи и администраторы, работающие с программным продуктом должны пройти обязательное обучение в компании ООО «Инженико» или компании-партнере и получить сертификат.

При возникновении вопросов по работе с программным обеспечением, обращайтесь в службу поддержки ООО «Инженико» - support@ingenico.ru.

Допускается незначительные несоответствия данной документации и программного обеспечения, связанные с постоянным развитием программных продуктов.

Не допускается использование текстов и изображений, входящих в данный документ, без согласования с ООО «Инженико».

¹ Так же допустимо использование названия «Arcom Universal EMV POS».

1_ARCUS 2.1 CAP

Arcus 2.1 CAP – это библиотека, предназначенная для минимизации необходимых изменений в существующих решениях кассового программного обеспечения (ПО) при интеграции терминального оборудования Ingenico на платформах Telium 1/2, Unicapt32 и ККМ, работающие под управлением данного ПО.

Библиотека является связующим звеном между кассовым ПО и терминальным оборудованием, реализуя обмен данными и обеспечивая диалог кассир-терминал на экране кассы или терминала (Режим работы с кассой).

ARCUS 2.1 CAP отвечает за:

- поддержку протокола обмена с POS-оборудованием через коммуникации ККМ;
- прозрачную передачу данных между POS-оборудованием и хостом банка;
- взаимодействие ПО POS-оборудования с кассиром в процессе транзакции по банковской карте.

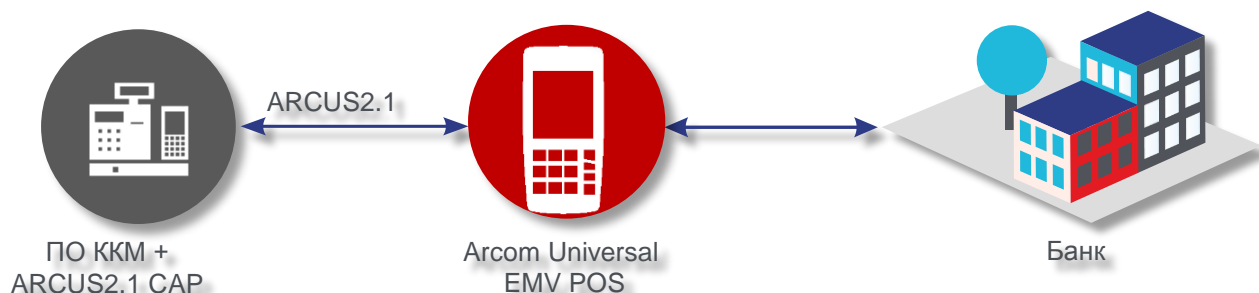


Рис. 1. Схема работы ARCUS 2.1 CAP

Программное решение работает на кассовом ПО в следующих операционных системах:

- Windows (XP SP3 и выше);
- Linux (в некоторых случаях может потребоваться перекомпиляция модуля под конкретный тип операционной системы);

1_1 Состав программы, основные принципы ее функционирования

Программное обеспечение решения Arcus2 поставляется в виде **модулей**:

EMV приложение на ПИН-пад или терминале Ingenico – (PAP²).

Приложение представляет собой стандартное POS приложение, модифицированное в части взаимодействия с кассовым ПО. Приложение ПИН-пад **полностью**³ управляет процессом выполнения операций: взаимодействием с процессингом, диалогом с кассиром и клиентом, подготовкой чеков, так как это выполняет обычное POS приложение автономного терминала.

Кассовое приложение (библиотека) — CAP⁴. (DOS/Windows/Linux)

Приложение CAP обеспечивает:

- прозрачную передачу данных между ПИН-падом и процессингом (напрямую по TCP/IP с использованием сетевого интерфейса кассового компьютера (режим «IP через кассу») если не предусмотрено отдельное подключение ПИН-пад в локальную сеть ТСП);
- взаимодействие (диалог) приложения ПИН-пад с кассиром;
- интеграцию с АРМ кассира.

Кассовое приложение (библиотека) ARCUS 2.1 CAP является зависимой от поставщиков АРМ кассира частью. Сегодня на рынке кассового ПО существует широкий список поставщиков,

2 Pin-pad Application

3 Предыдущая версия ARCUS управление ПИН-падом брала на себя

4 ARCUS 2.1 CAP - Cash Register Application

отличающихся между собой как платформой (Windows, Linux, UNIX, etc), используемым сетевым интерфейсом (TCP/IP), так и способностью к адаптации (взаимодействию) к ПО иных производителей. Поэтому модуль CAP – это набор реализаций одной задачи для разных поставщиков АРМ кассы. Наиболее распространенный вариант поставки кассового приложения CAP – Windows или Linux приложение⁵, взаимодействующее с ПИН-падом по RS232 или USB интерфейсу и с процессингом по TCP/IP.

1_1_1 Использование TLS шифрования в режиме IP через кассу



Рис. 2. TLS шифрование в режиме IP через кассу

Логика работы:

1. Данные о транзакции шифруются на терминале при помощи библиотеки TLS, и с использованием протокола ARCUS2 передаются в зашифрованном виде на ККМ.
2. ККМ принимает данные и, используя свои коммуникации, передает их на хост.
3. Хост расшифровывает полученные данные, проводит операцию, формирует ответ в зашифрованном виде и передает его ККМ.
4. ККМ получает ответ и при помощи протокола ARCUS2.1 передает его терминалу.
5. Терминал получает ответ от ККМ, расшифровывает его при помощи библиотеки TLS, и завершает операцию.

1_2 Режим работы с кассой

Платежный терминал может работать в двух режимах:

- автономный режим – терминал подключен к банковскому хосту через коммуникационный канал (например, Ethernet, GPRS), кассир инициализирует операцию и вводит данные для оплаты на терминале (производится двойной ввод данных), получает подтверждение проведения операции и вводит подтверждение оплаты на кассе;
- режим работы с кассой – терминал физически подключен к кассе (например, через USB, Bluetooth) и ожидает от нее команду и данные, необходимые для проведения операции, выполняет необходимые операции и возвращает кассу результат проведения транзакции.

Для переключения терминала в режим работы с кассой выполните следующие действия:

1. Для терминалов Telium при перезагрузке или включении питания, дождитесь появления окна с сообщением «Для настройки работы с кассой нажмите 'F'» и нажмите кнопку 'F'.




При определенных настройках для перехода в режим работы с кассой терминал может запросить пароль.

2. Для переключения терминала UNICAPT32 в режим работы с кассой удерживайте клавишу F3 (верхняя-правая кнопка, так же может называться <Menu> или на ней может быть изображен квадрат.) при включении питания или в процессе перезагрузки.
3. Работа с кассой?

- НЕТ – терминал загружается в автономном режиме;

⁵ Вариант для Linux


- ДА – терминал переходит к настройке режима работы с кассой.



В Arcom Universal EMV POS версии 4.0.2.435 и выше настройки режима работы с кассой производятся при помощи ARCUS меню.
Для упрощения процедуры заполнения параметров используется конфигурационный файл ARCUS.CFG, в котором задаются значения всех необходимых параметров. При последующем переключении в Режим работы с кассой, заданные вручную параметры сбрасываются в состояние, соответствующее настройкам файла ARCUS.CFG или значениям по-умолчанию, если файл отсутствует.
Описание ARCUS меню, см. Arcom Universal EMV POS – руководство администратора.

Файл ARCUS.CFG – текстовый файл в кодировке CP1251, содержит настройки и параметры подключения, загружается на терминал в раздел HOST, аналогично терминальному приложению (подробнее см. Arcom Universal EMV POS – Руководство по установке).

Для того, чтобы параметры из файла ARCUS.CFG применились выполните «5» - ARCUS 2 Настройки – Режим работы с кассой – ДА. На экране должно отобразиться сообщение «ПАРАМЕТРЫ КАССЫ ЗАГРУЖЕНЫ».




Параметры настройки и их значения отличаются, в зависимости от протокола.

1_2_1 Файл настроек ARCUS.CFG для Режимы работы с кассой терминального приложения ISO (Openway, SmartVista, Tieto, CTL)

/ Таблица 5. Описание файла настроек ARCUS.CFG ISO

Параметр	Описание	Возможное значение
Настройки интерфейса – раздел #Interface		
PORT	Порт подключения	COM0 - последовательный порт непосредственно на терминале USB – USB порт непосредственно на терминале COM_SL – последовательный интерфейс на MagicBox COMU – USB порт на базе ETH BLUETOOTH
PORT-SPEED	Скорость передачи данных	2400/9600/19200/115200
DATA-BITS	Количество бит данных	7/8
PARTITY	Контроль четности	NONE/EVEN/ODD
STOP-BITS	Стоп биты	1/2
ARCUSSEVER -IP	IP для подключения к ARCUS Net Server	Значение в формате xxx.xxx.xxx.xxx
ARCUSSEVER -PORT	Номер порта для подключения к ARCUS Net Server	Значение в формате xxxx
BLUETOOTH-PIN-LENGTH	Длина ПИН кода для подключения по Bluetooth	4-8
USE-ARCUS-ID	Использовать ARCUS ID	YES/NO
USE-TMSID	Использовать TMS ID	YES/NO

ARCUS-ID	Значение ARCUS ID	
Настройки сети – раздел #Network		
IP-ON-CASHREG	IP через кассу	YES/NO
STATIC-IP	Сетевые настройки терминала при работе с ARCUS NET Server	YES/NO
TERMINAL-IP		
MASK		
GATEWAY		
	Если значение параметра IP через кассу выставлено «NO», и работа осуществляется НЕ через NET сервер, сетевые настройки терминала задаются в коммуникационном скрипте! В этом меню их задавать НЕ нужно!	
Настройка таймаутов – раздел #Timeout		
DIALOG-TIMEOUT	Таймаут отображения диалогов	Значение задается в миллисекундах
DATA-TRANSMIT-TIMEOUT	Таймаут передачи данных	Значение задается в миллисекундах
CARD-READ-TIMEOUT	Таймаут чтения карты	Значение задается в миллисекундах
Настройки диалогов – раздел #Dialogs		
DIALOGS-ON-CASHREG	Отображать диалоги на экране кассы	YES/NO
Настройки маскирования номера карты при печати на чеке – раздел #SpecialMask		
SPECIAL-MASK	Активация доп. Маскирования	YES/NO
PAN-START	Количество отображаемых символов в начале номера карты	0-6, но не больше, чем указано в TMS
PAN-END	Количество отображаемых символов в конце номера карты	0-4 но не больше, чем указано в TMS
Настройки печати – раздел #PrintSettings		
PRINTER	Выбор принтера для печати чека	CASHREG/INTERNAL/BOTH
PRINTER-ENCODING	Кодировка печати	DEFAULT/CP866/CP1251
RECEIPT-WIDTH	Ширина чека в символах	24-48
FORMATTING	Разрешить спец. Символы	DON'T-USE-ESC/USE-ESC
Дополнительные настройки – раздел #AdditionalSettings		
SEND-PAN-SHA1	Передача SHA1 номера карты	YES/NO

Пример файла

```
#Interface
PORT=COM0
PORT-SPEED=2400
DATA-BITS=8
PARTITY=NONE
STOP-BITS=1
ARCUSSEVER -IP=192.168.0.101
ARCUSSEVER -PORT=9401
BLUETOOTH-PIN-LENGTH=4
USE-ARCUS-ID=YES
USE-TMSID=YES
ARCUS-ID=123456790

#Network
IP-ON-CASHREG=YES
STATIC-IP=YES
TERMINAL-IP=192.168.0.10
MASK=255.255.255.0
GATEWAY=192.168.0.1
```

```

#Timeouts
DIALOG-TIMEOUT=12000
DATA-TRANSMIT-TIMEOUT=1000
CARD-READ-TIMEOUT=6000

#Dialogs
DIALOGS-ON-CASHREG=YES

#SpecialMask
SPECIAL-MASK=YES
PAN-START=4
PAN-END=4

#PrintSettings
PRINTER=CASHREG
PRINTER-ENCODING=DEFAULT
RECEIPT-WIDTH=24
FORMATTING=DON'T-USE-ESC

#AdditionalSettings
SEND-PAN-SHA1=YES

```

Значения по-умолчанию

```


PORT=USB
PORT-SPEED=115200
DATA-BITS=8
PARTITY=NONE
STOP-BITS=1
IP-ON-CASHREG=YES (для iUN и по TE_02 - NO)
DIALOGS-ON-CASHREG=NO (для iUN и не SV_05 - YES)
DIALOG-TIMEOUT=12000
DATA-TRANSMIT-TIMEOUT=1000
CARD-READ-TIMEOUT=6000
PRINTER=CASHREG
PRINTER-ENCODING=DEFAULT
FORMATTING=DON'T-USE-ESC
SPECIAL-MASK=NO
SEND-PAN-SHA1=NO

```

1_2_2 Файл настроек ARCUS.CFG для Режимы работы с кассой терминального приложения Unipay (CompassPlus, SPDH)

/ Таблица 6. Описание файла настроек ARCUS.CFG UNIPAY

Параметр	Описание	Доступные значения
Настройки интерфейса – раздел #Interface		
PORT	Порт подключения	COM0/USB/COM_SL/COMU/ETH/BLUETOOTH
PORT-SPEED	Скорость передачи данных	2400/9600/19200/115200
ARCUSSEVER -IP	IP для подключения к ARCUS Net Server	Значение в формате xxx.xxx.xxx.xxx
ARCUSSEVER -PORT	Номер порта для подключения к ARCUS Net Server	Значение в формате xxxx
BLUETOOTH-PIN-LENGTH	Длина ПИН кода для подключения по Bluetooth	4-8
Настройки сети – раздел #Network		
IP-ON-CASHREG	IP через кассу	YES/NO
STATIC-IP	Сетевые настройки терминала при работе с ARCUS NET Server	YES/NO
TERMINAL-IP		
MASK		
GATEWAY		

Параметр	Описание	Доступные значения
 <p>Если значение параметра IP через кассу выставлено «NO», и работа осуществляется НЕ через NET сервер, сетевые настройки терминала задаются в коммуникационном скрипте! В этом меню их задавать НЕ нужно!</p>		
Настройки диалогов – раздел #Dialogs		
DIALOGS-ON-CASHREG	Отображать диалоги на экране кассы	YES/NO
Настройки маскирования номера карты при печати на чеке – раздел #SpecialMask		
SPECIAL-MASK	Активация доп. Маскирования	YES/NO
PAN-START	Количество отображаемых символов в начале номера карты	0-6, но не больше, чем указано в TMS
PAN-END	Количество отображаемых символов в конце номера карты	0-4 но не больше, чем указано в TMS
Настройки печати – раздел #PrintSettings		
PRINTER	Выбор принтера для печати чека	CASHREG/INTERNAL
RECEIPT-WIDTH	Ширина чека в символах	24-48
FORMATTING	Разрешить спец. Символы	DON'T-USE-ESC/USE-ESC
Дополнительные настройки – раздел #AdditionalSettings		
AUTH-CODE-FROM-CASHREGISTER	Получение кода авторизации с кассы	YES/NO
PRINT-JOURNAL	Настройки печати журнала операций	YES/NO/MISMATCH

Пример файла

```
#Interface
PORT=COM0
PORT-SPEED=19200
ARCUSSEVER -IP=192.168.0.101
ARCUSSEVER -PORT=9401
BLUETOOTH-PIN-LENGTH=4

#Network
IP-ON-CASHREG=YES
STATIC-IP=NO
TERMINAL-IP=192.168.0.10
MASK=255.255.255.0
GATEWAY=192.168.0.1

#Dialogs
DIALOGS-ON-CASHREG=YES

#SpecialMask
SPECIAL-MASK=YES
PAN-START=6
PAN-END=4

#PrintSettings
PRINTER=INTERNAL
RECEIPT-WIDTH=24
FORMATTING= USE-ESC

#AdditionalSettings
AUTH-CODE-FROM-CASHREGISTER=YES
PRINT-JOURNAL= YES
```

Значения по-умолчанию

```
PORT=COM0 (Если на терминале нет COM0, то USB (на UC32 - COM1))
PORT-SPEED=19200
RECEIPT-WIDTH=24
IP-ON-CASHREG=YES
DIALOGS-ON-CASHREG=YES
PRINTER=CASHREG
AUTH-CODE-FROM-CASHREGISTER=NO
SPECIAL-MASK=NO
PRINT-JOURNAL= NO
```

1_3 Интеграция с ККМ

1. При проведении безналичного платежа, кассовое Программное обеспечение (ПО) формирует данные по платежу. Данные могут различаться.

- Обязательные параметры:
 - код операции;
 - сумма операции в минимальных единицах валюты;
 - код валюты;



Начиная с версии, выпущенной после ноября 2014 г., транзакция прерывается, если с кассы не была передана сумма операции

- Необязательные параметры:
 - код авторизации [завершение расчета по предавторизации];
 - код ссылки RRN [завершение расчета по предавторизации].

В ряде случаев часть параметров может не передаваться с кассы, но тогда их необходимо будет вводить на терминале вручную.

2. Порядок вызова функций:
- a. Create – возвращается объект, с которым производятся дальнейшие действия.
 - b. Set – устанавливается необходимое значение: сумма, код валюты и т.д.
 - c. Run – запускается операция на POS терминале.
 - d. Get – получение данных, после завершения работы POS терминала.

3. В ходе выполнения безналичной операции по банковской карте из ПО ПИН-пад может быть затребован ввод дополнительных данных:

- код CVC2/CVV2
- PIN код
- Код авторизации
- RRN

Так же могут быть инициированы другие диалоги, связанные с подтверждением/отказом или выбором параметров операции из списка.




**Интерфейс ведения диалогов определяется настройками модуля ARCUS2 и ПО ПИН-пад и может быть назначен на использование клавиатуры и дисплея ККМ или ПИН-пад.
Вся текстовая информация передается в кодировке, зависящей от ОС ККМ (WIN1251 для Windows, DOS866 для DOS и Linux).**

4. Возможны следующие варианты диалогов с кассиром и держателем карты:

- Вопросы, на которые необходимо ответить Да/Нет;
- Выбор из списка значений;
- Приглашение на ввод текстовой, или числовой информации;
- А также информационные сообщения и сообщения об ошибках.

5. Возвращаемым значением функции **Run ()**, является код ответа от сервера (response code). При положительном результате выполнения операции возвращается «000», все остальные значения указывают на то, что операция не выполнена.



Коды ответов зависят от типа используемого процессинга. ARCUS2 так же использует внутренние коды ответов:

- **992=ОПЕРАЦИЯ ПРЕРВАНА КАССИРОМ ИЛИ КЛИЕНТОМ**
 - клиент или кассир отказались от проведения операции в момент ведения диалогов на ПИН-паде;
 - чиповая карта «отказалась» от проведения финансовой операции (напр. ответила кодом AAC);
 - в параметрах вызова модуля ARCUS от кассовой программы получены недопустимые параметры (например, несуществующий в параметрах терминала КОД ВАЛЮТЫ)
- **998=ОШИБКА СВЯЗИ, ПОЗВОНИТЕ В БАНК - попытка IP-соединения ПИН -пада с сервером авторизации завершилась:**
 - ошибкой;
 - превышением тайм-аутов;
 - обрывом соединения и пр.
- **999=НЕТ СВЯЗИ С ПИН-ПАДОМ - проблема обмена данными по RS232/USB интерфейсу между ПИН-падом и модулем ARCUS на кассе:**
 - обрыв проводов или неисправность разъемов
 - проблема с питанием ПИН-пада
 - USB ПИН-пад по инициативе кассового ПК перешел в режим энергосбережения (определяется настройками управления питанием в BIOS компьютера, моделью используемого USB кабеля для подключения ПИН-пада к ПК)

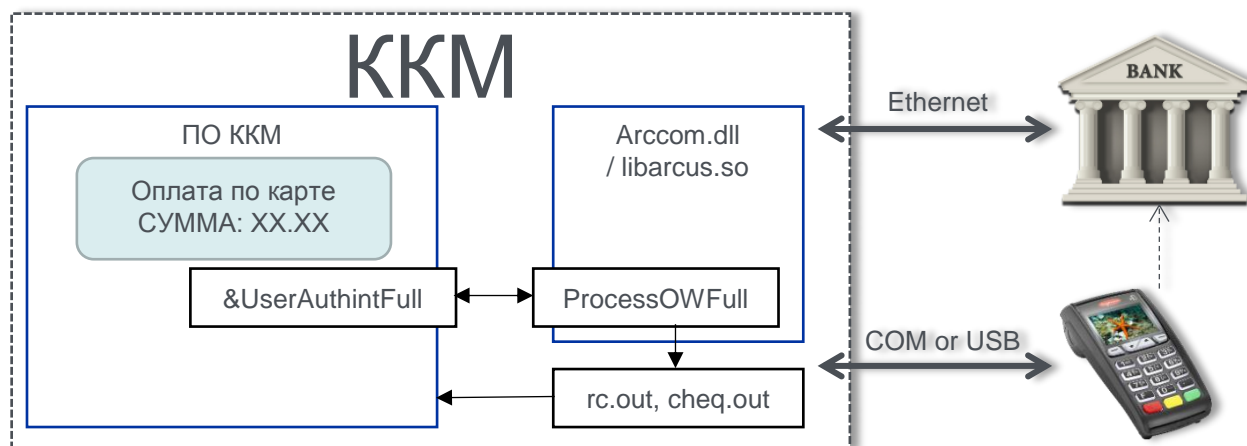


Рис. 3 Схема взаимодействия кассового ПО с модулем Arcus2.1 с использованием Arccom.DLL

Результатом работы модуля ARCUS2.1 являются формирование двух (трех) текстовых файлов:

- Файл кода ответа (например, RESULT_FILE или rc.out), содержащий трехзначный код ответа от банка. При положительном результате выполнения операции возвращается «000», все остальные значения указывают на то, что операция не выполнена (в операции отказано).
- Файл - чек (например, cheq.out), содержащий данные по операции, для печати на принтере в кодировке, соответствующей ОС. Формат и содержание информации в файле чека банковской операции определяется настройками специалистов Банка при параметризации ПО терминала. Файл чека содержит только печатные символы и не управления форматом печати и ESC-последовательностей.

- Файл данных [опция] (имя файла для ОС Windows и DOS output.dat, для Linux output.out), содержащий строки с данными по операции, в кодировке, соответствующей ОС (файл текстовый). Назначение и формат данных по строкам фиксированный (см. п. 1_3_1 Формат и содержание файла данных output.dat).

Пример

В случае проведения ARCUS2 безналичной транзакции в файле чека будет содержаться банковский чек.

```

=====пример= чека=====
    ОАО "Банк ВТБ
    Северо-Запад"
    пр. М. Говорова, д.52
    Банк ВТБ Северо-Запад
    Терминал   R0000009
    16/07/10 13:29  Кассир:0

    Оплата Покупки
    0.02 RUR
    Код ответа:      1
    Код авторизации 404365 В
    Карта 4*****6016
    VISA      01/11
    CARD TEST
    Чек #114   1.1.4
    Операцию подтверждаю:
    _____
    (Подпись клиента)
=====конец примера=====
  
```

Рис. 4. Банковский чек

При возникновении программно-аппаратного сбоя в файле кода ответа будет содержаться код ошибки, а в файле чека технический чек с кодом и расшифровкой:

```

=====пример= чека=====
    ОШИБКА
    Код ошибки:999 (999)
    Нет кода ответа
    =====конец примера=====

    =====пример=====
    999
    =====конец примера=====
  
```

Рис. 5. Чек с ошибкой

1_3_1 Формат и содержание файла данных output.dat

```

000          - код ответа (NNN 000 - 999)
447520*****7014 - PAN
R0000009    - Terminal ID (идентификатор терминала в платежной системе)
738593      - код авторизации (для офлайн операций - последние 6 цифр RRN)
VISA        - название типа карты
ОДОБРЕНО    - текстовое сообщение
500         - сумма транзакции (если была указана в командной строке) 5.00р.
12          - номер чека в терминале
4816D0C8FF699AE1B7638672E880D7943DC95117598971170C1C9D921F9AF09C
              - криптограмма данных о карте
  
```

В результате транзакции терминал формирует чек, который можно распечатать на внутреннем принтере терминала или на принтере чеков, подключенном к кассе.

Формат чека при печати на встроенном принтере определяется настройками специалистов Банка при параметризации ПО терминала.

В случае если для печати используется принтер, подключенный к кассе, то данные для печати можно получить несколькими способами:

- напечатать содержимое файла-чека **cheq.out** и вывести на экран данные из файла ответа **rc.out**;
- кассовое ПО самостоятельно формирует чек безналичной операции на основании данных, полученных из вызова Get.

Кассовое ПО самостоятельно формирует чек безналичной операции на основании результирующего файла данных **output.dat**. Обязательный набор данных на кассовом чеке, при проведении операции по банковской карте:

- Код авторизации;
- Валюта операции;
- Место для подписи клиента;
- Наименование, страна и город (адрес) торгового предприятия;
- Код ответа и расшифровка причины отказа;
- Тип карты и дата истечения карты;
- Наименование обслуживающего банка;
- EMV данные (при проведении операции по чиповой карте).

В случае использования в ПО ККМ файла чека, выполнение требований международных платежных систем по наличию и содержанию информации на чеке операции по карте осуществляется специалистами Банка при параметризации ПИН-пад.

В случае использования в ПО ККМ данных от модуля ARCUS 2.1 или файла данных выполнение требований международных платежных систем по наличию и содержанию информации на чеке операции по карте должно выполнять ПО ККМ.

Исключение, не требующее печати чека, составляют коды 99x (x = 1...9) — коды технологических сбоев или отказов, сгенерированные ARCUS 2.1, которые не являются кодами ответа сервера Банка по безналичной операции.

1_4 Вызов ARCUS 2.1

1_4_1 Windows / Linux

Стандартным способом интеграции с ККМ является использование библиотеки Arcscom.dll / libarcus.so (см. раздел Интеграция с ККМ).

Для проверки работы модуль ARCUS2 поставляется вместе с эмулятором кассовой программы **CommandLineTool**. Данная утилита используется для полноценной проверки работы терминала, подключенного к ПК с ОС Windows.

- **/o[Код операции ККМ]** – Оплата, Отмена, Возврат и т.д. Цифровое значение соответствует коду операции в файле «ops.ini»;
- **/c[Код валюты]** – код валюты операции;
- **/a[Сумма]** – сумма операции в минимальных единицах валюты без точки;
- **/i[Идентификатор терминала]** - ID терминала до 9 символов (определяется TMS);
- **/v[Код авторизации]** - код авторизации, 6-9 символов;
- **/r[Ссылка]** — RRN, до 12 символов;
- **/e[Данные карты]** - шифрованные данные о карте, 64 символа;
- **/t[Трек 2]** – второй трек пластиковой карты;
- **/b[TraceID]** – ID предавторизации (доступно с января 2015г. TE_02);
- **/h[Original Amount]** – общая авторизованная сумма (доступно с января 2015г. только для продукта TE_02);
- **/f[Original Date/Time]** – дата и время оригинальной Предавторизации (доступно с января 2015г. только для продукта TE_02);
- **/gb** – проверка работы сканера штрих кодов терминала iSMP (доступно с января 2015 г. Только для продукта CTL_03);

- **/pd** – данные для проведения платежа в сторону третьих лиц (доступно с февраля 2015г. только для продукта с кодом TPTP_04);
- **/grN** – вызов печати дубликата чека. N – Номер чека (доступно с версии ПО 4.0.2.407 для продукта SV_05).
- **/dlsh** – передача дополнительных данных от кассы (только для продуктов с функционалом DLS).
- **--goods-file=[полный_путь_и_имя_файла]** – путь к файлу со списком товаров (с июля 2016 года в ARCUS2 CAP доступна функция передачи информации о товаре).
- **--bonus=NNNN** - бонусы для списания в минимальных единицах валюты (1000 = 10 бонусов).
- **--card-type=1** – признак типа карты (1 - банк, 2 - топливо, 4 - лояльность).
- **--no-icc-removing** выполняет команду с кассы без извлечения карты, если использовалась чиповая карта.



Обязательными параметрами для передачи являются: Код операции, валюта и сумма. Другие параметры могут отсутствовать. В случае необходимости терминал запросит недостающую информацию (например, номер RRN (Retrieval Reference Number) для операции «Отмена»). Кассир вводит данные по приглашению на экране терминала или кассы.



Специфика работы unattended-устройств (iUN250, i9500 и др.) не подразумевает ввод дополнительной информации клиентом, поэтому при работе с такими терминалами необходимо передавать все необходимые параметры, например, «Код валюты» и «Сумму» при операции «Оплата».

Пример использования:

- Оплата на 123,45 рублей, код валюты 643.

```
CommandLineTool.exe /o1 /c643 /a12345
```



Номер карты и срок действия может быть введен вручную с клавиатуры ПИН-пад. Для этого в момент приглашения на экране ПИН-пад «Вставьте карту» необходимо нажать желтую кнопку «С».

- Передача данных трек 2, например, при чтении магнитной полосы на клавиатуре кассы. Оплата на 123.45 рублей, код валюты 643, трек карты «1000023000000034=06062011021300000641».

```
CommandLineTool.exe /o1 /c643 /a12345 /t;1000023000000034=06062011021300000641?
```

Данные трек 2 предваряются символом «;» и завершаются «?».

В случае эмуляции ручного ввода данные передаются следующим образом:

```
M[Номер карты]=[срок действия в формате ГГММ]?
CommandLineTool.exe /o1 /c643 /a12345 /tm1000023000000034=1106?
```



ARCUS 2.1 исключен из области сертификации Payment Application Data Security Standards (PA DSS), так как при правильной настройке не хранит и не передает данные платежных карт в нешифрованном виде.



При использовании режима HRS, для проверки зашифрованных данных о карте, зашифрованные данные передаются после ключа «/e».

Пример запуска:

```
CommandLineTool.exe /o1 /a2000  
/e3AB5E25D6091053EAB71AFAE303B5327F3D713043819649F3D20D11F78A73811
```

Получение зашифрованных данных настраивается в меню администратора - доп. настройки – РЕЖИМ HRS (использовать шифрование параметров).

Данные шифруются с помощью 3DES CBC. Ключ для шифрования вводится в открытом виде в меню администратора – ПИН-пад - ARCUS КЛЮЧ.

1_5 Настройки ARCUS 2.1 CAP

1_5_1 Файл инициализации Cashreg.ini

Файл инициализации входит в установочный комплект ARCUS2 и подробно описан в Руководстве по установке ARCUS 2.

1_5_2 Файл ops.ini для ISO8583

Файл соответствия кода операции, передаваемого с ККМ и кода операции ПИН-пад (пример для протоколов на базе ISO8583 - OpenWay, SmartVista, Tieto).

/ Настройки операции:

Op code	Op class	Op id	Op Name
1 =	1,	1,	ОПЛАТА

- **Op code** - код операции для кассы
- **Op class** - класс операции
- **Op id** - ID операции

/ Таблица 7. Список операций для интеграции с ПО Arcom Universal EMV POS (NewWay)

Класс	ID	Название
ОСНОВНЫЕ ФИНАНСОВЫЕ ОПЕРАЦИИ		
1	1	ОПЛАТА ПОКУПКИ / ОПЛАТА ТОВАРА / ОПЛАТА
1	2	ОПЛАТА ПОКУПКИ DUAL
1	3	ПРЕДАВТОРИЗАЦИЯ
1	4	ЗАВЕРШЕНИЕ ПРЕДАВТОРИЗАЦИИ / ЗАВЕРШЕНИЕ РАСЧЕТА
1	5	УНИВЕРСАЛЬНАЯ ОТМЕНА
1	6	ОТМЕНА ADVICE (TIETO)
1	8	МИНИ-ВЫПИСКА
1	10	УНИВЕРСАЛЬНАЯ ОТМЕНА ADVICE
1	11	ВОЗВРАТ / ВОЗВРАТ ТОВАРА
1	15	CREDIT VOUCHER
БАНК, ПУНКТ ВЫДАЧИ НАЛИЧНЫХ		
1	7	ЗАПРОС БАЛАНСА (в т.ч. для АДМ и работы с депозитными машинами SV_05)
1	12	ВЫДАЧА НАЛИЧНЫХ (в т.ч. для АДМ)

Класс	ID	Название
1	13	ВЫДАЧА НАЛИЧНЫХ DUAL
1	14	КРЕДИТ СЧЕТА КАРТЫ (в т.ч. для АДМ и работы с депозитными машинами SV_05)
1	47	BALANCE ACC
ПЕРЕВОДЫ, ПЛАТЕЖИ		
1	16	ПЛАТЕЖ UTILITY_PAYMENT
1	17	ПЛАТЕЖ BILLING_PAYMENT_SMS
1	18	ПЛАТЕЖ BILLING_PAYMENT_DUAL
1	40	ПЕРЕВОД Card2Card
1	41	ПЕРЕВОД Cash2Card
ТЕХНИЧЕСКИЕ ОПЕРАЦИИ		
2	3	АВТООТМЕНА (ТЕХНИЧЕСКАЯ ОТМЕНА)
2	4	ЧТЕНИЕ МАГНИТНОЙ КАРТЫ
2	6	ТЕСТ СВЯЗИ С ХОСТОМ проверяется только возможность соединения с ним, без авторизации на хосте, отправки ему команды и печати чека
2	18	ТЕСТ СВЯЗИ EXCHANGE (ЗАПРОС - ОТВЕТ) тестирует связь с хостом (авторизуется, отправляет echo запрос), после чего печатает чек (SV).
2	21	ИНФО ТЕРМИНАЛА
ОТЧЕТЫ / ИТОГИ		
2	0	ПОЛНЫЙ ЖУРНАЛ
2	1	СВЕРКА ИТОГОВ
2	10	КРАТКИЙ ОТЧЕТ
2	19	ОТПРАВКА ОТЛОЖЕННОЙ ОТМЕНЫ / СООБЩЕНИЯ (SV)
ПЕЧАТЬ ДУБЛИКАТА ЧЕКА		
2	27	ПЕЧАТЬ ДУБЛИКАТА ПОСЛЕДНЕГО ЧЕКА
РАБОТА С МЕНЮ		
2	2	МЕНЮ КАССИРА
2	9	ПАРОЛЬ СТАРШЕГО КАССИРА
КОМАНДЫ ДЛЯ РАБОТЫ В NEWWAY С ПРОЦЕССИНГАМИ SPDH / COMPASS+		
2	0	ОТЧЕТ (ПАКЕТ)
2	23	ОТЧЕТ (СМЕНА)
2	24	ОТЧЕТ (ДЕНЬ)
2	1	СВЕРКА ИТОГОВ (ПАКЕТ)
2	25	СВЕРКА ИТОГОВ (СМЕНА)
2	22	СВЕРКА ИТОГОВ (ДЕНЬ)
АДМИНИСТРАТИВНЫЕ ОПЕРАЦИИ		
3	1	СИНХРОНИЗАЦИЯ ВРЕМЕНИ
3	2	СИНХРОНИЗАЦИЯ ВРЕМЕНИ
3	4	СЕССИЯ TMS
3	5	МЕНЮ АДМИНИСТРАТОРА


Класс	ID	Название
ОПЕРАЦИИ С НАЛИЧНЫМИ (в TMS должен быть добавлен тип карт «Наличные»)		
4	89	ОПЛАТА НАЛ
4	90	ВОЗВРАТ НАЛ



ПО Universal EMV POS поддерживает функционал Динамического меню – в Режиме Работы с кассой для пунктов меню используется класс команд «8», в качестве Op id необходимо использовать значение, указанное в TMS – Терминальные группы – Динамическое меню – Касса (может принимать значения от 1 до 20).

1_5_3 Файл ops.ini для Compass+ и SPDH

Класс	ID	Название
ОСНОВНЫЕ ФИНАНСОВЫЕ ОПЕРАЦИИ		
0	128	ОПЛАТА ПОКУПКИ (SPDH)
0	130	ВОЗВРАТ ТОВАРА (SPDH)
0	133	ПРЕДАВТОРИЗАЦИЯ
0	134	ЗАВЕРШЕНИЕ ПРЕДАВТОРИЗАЦИИ
0	224	ОТМЕНА ПОСЛЕДНЕЙ (ТЕХНИЧЕСКАЯ ОТМЕНА, АННУЛИРОВАНИЕ ЧЕКА)
0	225	ОТМЕНА ПО ЧЕКУ (УНИВЕРСАЛЬНАЯ)
СВЕРКА, ОТЧЕТЫ		
0	186	ПЕЧАТЬ ЖУРНАЛА ТРАНЗАКЦИЙ
0	187	ПОЛНЫЙ ЖУРНАЛ
0	180	КРАТКИЙ ОТЧЕТ
0	160	ЗАКРЫТИЕ ПАКЕТА
0	161	ЗАКРЫТИЕ СМЕНЫ
0	162	ЗАКРЫТИЕ ДНЯ
0	181	ПЕЧАТЬ ЖУРНАЛА
0	179	ЛОКАЛЬНЫЕ ИТОГИ
БАНК - ПУНКТ ВЫДАЧИ НАЛИЧНЫХ		
0	132	ВЫДАЧА НАЛИЧНЫХ
0	131	ПРИЕМ НАЛИЧНЫХ
0	136	БАЛАНС
0	231	СМЕНА ПИН-КОДА
0	129	БЕЗНАЛИЧНЫЙ ПЕРЕВОД
0	154	ПРОВЕРКА КАРТЫ
КОРРЕКЦИИ		
0	144	КОРРЕКЦИЯ ПОКУПКИ
0	147	КОРРЕКЦИЯ ПОПОЛНЕНИЯ НАЛИЧНЫМИ
0	148	КОРРЕКЦИЯ ВЫДАЧИ
ПЛАТЕЖИ ПЕРЕВОДЫ		

Класс	ID	Название
0	138	ПЛАТЕЖ В СТОРОНУ ТРЕТЬИХ ЛИЦ (ТРТР_04)
0	142	ПЕРЕВОД С КАРТЫ НА КАРТУ
0	143	ПОПОЛНЕНИЕ КАРТЫ НАЛИЧНЫМИ
СЕРВИСНЫЕ		
0	176	КОПИЯ ПОСЛЕДНЕГО ЧЕКА
0	177	КОПИЯ ЧЕКА ПО НОМЕРУ
0	182	ДЕТАЛИ EMV
СПЕЦИАЛЬНЫЕ ВЕРСИИ ПО		
0	220	<p>ОТМЕНА ПО СУММЕ, сумма - обязательный параметр (SPDH_03)</p>  <p>Данный вид отмены доступен только в режиме работы с кассой, при включенном параметре на TMS "требовать карту для отмены"</p>
АДМИНИСТРАТИВНЫЕ ОПЕРАЦИИ И МЕНЮ КАССИРОВ		
0	166	ТЕСТ СВЯЗИ С ХОСТОМ
0	167	ЗАГРУЗКА ПАРАМЕТРОВ С ХОСТА (SPDH)
0	50	МЕНЮ КАССИРА
0	53	АДМИНИСТРАТИВНОЕ МЕНЮ
0	204	СЕССИЯ TMS

1_5_4 Файл rc_conv.ini

В файле rc_conv.ini (фрагмент) заполняется таблица преобразования кода ответа устройства ПИН-пад в код ответа необходимый для ККМ.

К примеру, если необходимо чтобы код ответ **001** записывался как **005**, а код ответа **123** как **843**, то для этого в файл rc_conv.ini необходимо добавить строки в следующем формате:

[Код ответа ПИН-пад]=[Код ответа ККМ].

Например:

001=005
123=843

Добавление этих строк означает что, в случае если ПИН-пад присылает код ответа «**001**», то код ответа будет заменен на «**005**», если же код ответа «**123**», то в файл ответа будет записано «**843**».

В случае если в файле rc_conv.ini данный код ответа отсутствует, то будет использовано значение, полученное с ПИН-пад.



Некорректные данные в файле rc_conv.ini могут привести к неверной обработке ККМ кодов ответов терминала и последующим финансовым последствиям. Разработчики и администраторы ПО ККМ должны предпринять меры по защите файла rc_conv.ini от несанкционированных изменений

1_5_5 Файл rc_res.ini

В файле Rc_res.ini (фрагмент) задаются коды ответа процессинга.

В файле заполняется таблица соответствий кодов ответа и информационных сообщений, выводимых на экран. Строки файла имеют следующий формат:

[Код ответа]=[Текстовое сообщение]

000= Одобрено
001= Одобрено
002= Одобрено
003= Одобрено
004= Одобрено
005= Одобрено
006= Одобрено
007= Одобрено
008= Одобрено
009= Одобрено
050= Отказ
051= Карта просрочена
052= Лимит PIN исчерпан
053= Чужая адм. карта
054= Нет блока секретности
055= Неверная операция



Подробная информация по содержимому файлов инициализации находится в именуемом у Вас ТЗ на разработку или внедрению проекта.



При работе с ОС Linux регулируется флагом USEORIGINALRC в файле cashreg.ini (см. 1.3.3 Файл Cashreg.ini).

1_6 Эксплуатация

1_6_1 Интеграция ARCUS 2.1 CAP с ПО ККМ

ARCUS 2.1 CAP в результате доработок кассового ПО становится дополнительным модулем ПО ККМ, расширяющим возможности функционирования ПО ККМ в плане выполнения расчетов по банковским картам при проведении операций оплаты товаров и услуг в торговых и сервисных предприятиях и осуществления операций со счетами клиентов в отделениях Банков.

1_6_2 Виртуальная клавиатура

Виртуальная клавиатура доступна для касс, работающих на ОС Windows, с сенсорным экраном с июля 2014 года для ПО Unipay. Клавиатура не отключаемая и для её работы необходима специальная версия dialogs.dll.



Рис. 6. Виртуальная клавиатура

При помощи виртуальной клавиатуры возможно осуществлять ввод следующих данных:

- цифровой ввод – редактирование;
- ввод/редактирование IP- адресов;
- ввод/редактирование HEX.

1_6_3 Способы и периодичность контроля правильности работы ПО

Контроль за правильностью работы ARCUS 2.1 CAP осуществляется в рамках комплексного технического обслуживания эквайринга Банка в ТСП, техническими специалистами служб эквайринга банковских карт со стороны процессинга и фирмы производителя ARCUS 2 CAP на основании статистической информации по сбоям, которая поступает из процессинга Банка и от IT-специалистов ТСП.

В случае получения уведомления от производителя кассового ПО о выходе обновлений существующей версии кассового ПО или выпуске принципиально новой версии кассового ПО, осуществляются мероприятия по контролю совместимости настроек модуля ARCUS 2.1 CAP и выполнению тестовых испытаний на стенде кассовой компании или в ТСП.

В случае выхода обновлений существующей версии ARCUS 2.1 CAP или выпуске принципиально новой версии ARCUS 2.1 CAP, осуществляются мероприятия по контролю совместимости интерфейса и настроек модуля ARCUS 2.1 CAP и выполнению тестовых испытаний на стенде кассовой компании или в ТСП.

1_6_4 Аварийные ситуации и способы их устранения

При возникновении аварийных ситуаций (затирания или удаления критических файлов из состава ARCUS 2.1 CAP, краха ПО или ОС) техническими специалистами ТСП и службы тех. обслуживания эквайринга, осуществляются мероприятия по восстановлению и настройке модуля ARCUS 2.1 CAP и выполнению пусковых испытаний в ТСП.

При возникновении аварийных ситуаций (сбоев при выполнении финансовых и административных операций по банковским картам) техническими специалистами ТСП и службы тех. обслуживания эквайринга, осуществляются мероприятия по анализу причин сбоев на основании данных из файлов логирования ARCUS 2.1 CAP, составлению отчетов по выявленным причинам и передача информации в службы поддержки программных продуктов и департамент разработки ПО ARCUS 2.1 CAP для проведения работ по устранению ошибок ПО ARCUS 2.1 CAP и выпуску обновления, после чего происходит выполнение отладочных, тестовых и пусковых испытаний в службах тестирования производителя ARCUS 2.1 CAP и в ТСП.

1_6_5 Описание фрагментов файла лога

- Версия сборки Arccom.dll

Version: 2.1.0.1
Build date:
Tue Dec 23 17:56:01 2014

1_6_6 Возврат без предъявления карты

Данный функционал доступен для продукта SPHD_03.

Функционал реализует возможность проведения операций «Отмена», «Возврат» и «Завершения предавторизации» по коду авторизации без карты клиента в режиме работы с кассой.

Так как в журнале операций хранится только номер карты и срок действия, операции «Возврат» и «Завершения предавторизации» по коду авторизации выполняются только в режиме ручного ввода (без данных треков и данных EMV).

Для операций «Отмена», «Возврат» и «Завершения предавторизации» в режиме работы с кассой терминал обрабатывает полученные с кассы код авторизации, сумму и 4 последних цифры номера карты⁶.

⁶ Терминал должен быть настроен для передачи на кассу как минимум 4 последних цифр номера карты в открытом виде

Для операции «Отмена» поиск выполняется по трем параметрам (код авторизации, сумма, 4 последних цифры номера карты). Поиск производится по текущему бизнес-дню.

Для операций «Возврат» и «Завершения предавторизации» поиск выполняется только по коду авторизации и 4 последним цифрам номера карты, а полученная с кассы сумма используется при проведении новой операции. Поиск производится по всему журнал транзакций в памяти терминала, включая операции по бизнес-дням, по которым уже было выполнено закрытие дня.

Если операция в журнале найдена, то терминал выполняет операцию без необходимости ввода карты клиента.

Если операция не найдена, то терминал продолжает работу по стандартному сценарию: запросить карту клиента (для операций «Возврат» и «Завершение расчета») или номер чека (для операции «Отмена»).

Если один или несколько параметров (код авторизации, сумма, 4 последних цифры номера карты) не были получены терминалом, то поиск транзакции выполняется без них (например, по коду авторизации и номеру карты или по коду авторизации и сумме).

Для операций «Возврат» и «Завершение предавторизации» терминал использует сумму, полученную с кассы. Операция «Отмена» производится только на полную сумму.

Для реализации функционала разработана специальная версия DLL.

Для передачи кода авторизации используется конструкция вида /vXXXXXX_X (пробел заменяется подчеркиванием).

- Для интеграции через CommandLineTool при передаче пробел заменяется подчеркиванием /vXXXXXX_X
- Для интеграции через библиотеку код авторизации записывается в поле "UserAuthIntFull.authCode" без преобразований.

Для передачи 4-х последних цифр номера карты используется конструкция вида /tDXXXX:

- Для интеграции через CommandLineTool (XXXX - последние 4 цифры номера карты) /tDXXXX
- Для интеграции через библиотеку в поле "UserAuthIntFull.pan" записывается значение вида (XXXX - последние 4 цифры номера карты): DXXXX

Пример команды Arcus2:

```
CommandLineTool.exe /o6 /a22222 /tD6416 /v669993_v
```

2_Интеграция через RS232 или USB

В зависимости от модели используемого терминала его подключение к ККМ может осуществляться либо через последовательный порт RS232, либо к USB порту.

В случае, если сетевое подключение к банковскому процессингу предполагается осуществлять через оборудование ККМ, то на ней должен быть установлен сетевой интерфейс, через который обеспечивается доступ в сеть. Подключение к процессингу возможно также без участия ККМ, а непосредственно от терминала, в таком случае необходимо наличие канала связи, к которому будет подключен терминал.

На ККМ должна быть установлена операционная система Windows 98/2000/XP, кассовое ПО, а также пакет ПО Arcus II.

При подключении терминала через порт USB на ККМ должен быть установлен драйвер виртуального COM-порта.

В подключаемый терминал должно быть загружено программное обеспечение, версия которого зависит от типа банковского процессинга, а также конфигурационные файлы.

2_1 Схема взаимодействия с ПО ККМ

При проведении безналичного платежа, кассовое ПО должно сформировать данные по платежу:

- Код операции;
- Код валюты;
- Сумма операции в сотых долях.

Через метод «Set», заполнить эти данные. Затем ПО кассы должно вызвать библиотечную функцию «Run», в качестве параметра передав ей номер операции. Вызов функции «Run» является блокирующим, управление диалогом с кассиром передается приложению ПИН-пад.

В ходе выполнения операции могут быть затребованы ввод кода CVC2/CVV2, ввод ПИН-кода. Вся текстовая информация передается в кодировке WIN1251.

3_Низкоуровневая интеграция

ПО ККМ может взаимодействовать с ПИН-пад без использования библиотеки ARCUS2. В этом случае в ПО ККМ должна быть реализована полная поддержка протокола ARCUS2.

ПИН-пад должен быть настроен на работу с ККМ в соответствии с документом «ARCUS2 для интеграции с кассовыми решениями - Руководство по установке, п. 3.2. Переключение терминала в режим работы с кассой». Устройство (ПИН-пад), после включения питания и запуска ПО находится в состоянии ожидания команды начала проведения операции по настроенному каналу связи (RS232, USB, TCP ...).

После получения команды ПИН-пад переходит в активный режим и начинает обработку запрошенной операции. В процессе выполнения устройство отправляет на ККМ команды открытия тех или иных ресурсов, чтения и записи, печати и отображения различных диалогов. Присланная команда должна быть обработана на ККМ, после чего должен быть выработан ответ на команду и отправлен на ПИН-пад.

В конце каждой операции ПИН-пад отправляет команду завершения операции, после ответа на которую, операция считается завершенной и устройство ККМ должно закрыть все открытые в процессе ресурсы. Операция считается завершенной.

Результат выполнения определяется по значению кода ответа присланного специальной командой с ПИН-пад-а. Отсутствие в процессе выполнения операции подобной команды считается отрицательным результатом выполнения (операция завершилась с неизвестной ошибкой). Так же в процессе проведения может быть прислано несколько таких команд, в этом случае в качестве значения кода ответа используется последнее присланное ПИН-пад-ом.



Рис. 7. Общая схема взаимодействия

Описание низкоуровневого протокола взаимодействия предоставляется по запросу в службу технической поддержки Ingenico.

4_Arcus NET Server

Arcus NET Server обеспечивает взаимодействие между платежными терминалами и ККМ, которое может быть организовано в двух вариантах:

- v.1.3 – только через TCP/IP;
- v.2.0 – через TCP/IP и (или) файловый обмен.



С октября 2014 года при внедрении новых проектов используется Arcus NET Server v.2.

Информация о версии Arcus NET Server отображается в логе при старте службы. По умолчанию лог расположен в директории "C:\ArcusNETServer\logs".

- v.2.0: "INFO r.i.s.ServerApp [main] Arcus NET Server (Version: 2.0 BETA Rev. 363) by Ingenico LLC."
- v.1.3: "INFO [main] - Initializing Arcus IP Server v1.3alpha"

Arcus NET Server это отдельный программный продукт. Один сервер может обслуживать до 50 ККМ-ПИН-пад⁷.

4_1 Arcus NET Server v.1.3

Arcus NET Server v.1.3 выступает в качестве промежуточного сервера, необходимого для соблюдения требований PA-DSS, запрещающих терминалу принимать входящие TCP/IP соединения.

Может быть использован в следующих ситуациях:

- для интеграций с ККМ, реализованных по схеме «тонкий клиент»;
- для случаев, когда необходимо подключить ПИН-пад к ККМ при помощи Ethernet (если их невозможно подключить напрямую по USB или COM).

Приложение может быть установлено на терминальный сервер, или на отдельный сервер. В обоих случаях на сервер должен быть установлен Arcus 2 CAP.

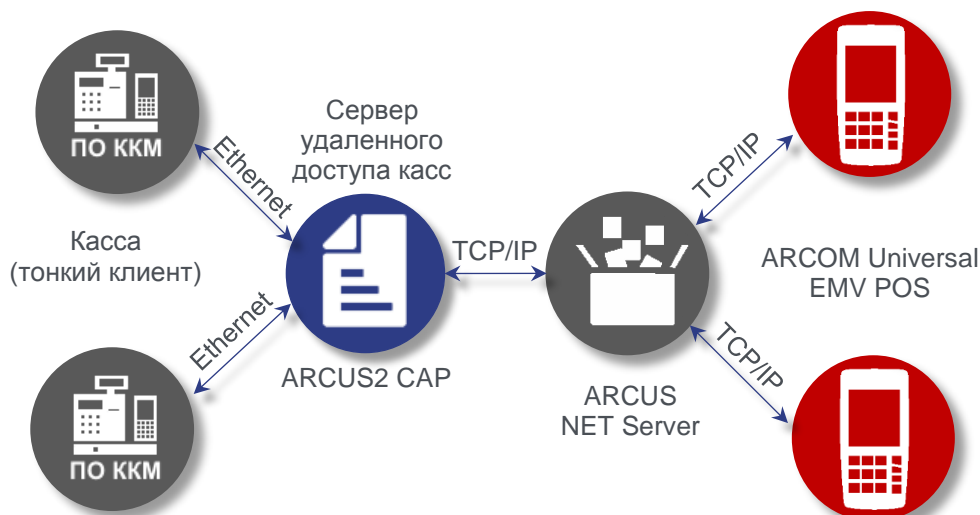


Рис. 8. Схема работы ARCUS NET Server v.1.3

При использовании Arcus NET Server существует несколько схем работы:

- одна ККМ, один ПИН-пад;

⁷ Указано количество для базовой лицензии. Для увеличения числа касс потребуются разработка мер административного характера и дополнительное нагрузочное тестирование

- одна ККМ, несколько ПИН-пад;
- несколько ККМ, один ПИН-пад;
- несколько ККМ, несколько ПИН-пад.

Все подключения «ККМ <-> ПИН-пад» происходят по Ethernet. Управление логикой подключений происходит с помощью формирования очередей (файлы pinpads.conf, cashregisters.conf, server.conf).

Для установки и настройки программ, необходимо последовательно решить несколько задач:

- установка Arcus2 на сервер удаленного доступа касс;
- установка Arcus NET Server и его настройка;
- подключение устройства ПИН-пад, настройка на работу с кассой.



Сервер Arcus NET Server v.1.3 написан на Java и ТРЕБУЕТ JRE не ниже 1.5

4_1_1 Файл Server.conf

Основной конфигурационный файл содержит следующие настройки:

- **server.interval=500** - Интервал в миллисекундах, с которым сервер обновляет свое состояние. Чем меньше - тем больше нагрузка на процессор, чем больше, тем он медленнее (дольше останавливается, например);
- **server.pinpads.pooltype=stack** - Стратегии подбора свободных устройств ПИН-пад из очереди:
 - **none** - одна очередь максимум один свободный ПИН-пад;
 - **queue** - Первый соединившийся свободный ПИН-пад - первым начинает обслуживание пришедшей транзакции;
 - **stack** - Последний соединившийся свободный ПИН-пад - первым начинает обслуживание пришедшей транзакции;



Если ПИН-пад, часто сам по себе перегружается или незапланированно рвет соединение с сервером, замените значение stack.

- **server.pinpads.idle=10000** - ПИН-пад раз в 1 секунду посылает "ping" на сервер, подтверждает, что он готов к работе. Значение параметра - время после которого считается, что ПИН-пад разъехался;
- **server.pinpads.binds=192.168.10.98:9301, 192.168.10.98:9302, 0.0.0.0:1234** – настройка входящих портов сервера для устройств ПИН-пад. Через запятую перечислены:
 - [IP адрес]:[PORT] с которого будут приниматься соединения ПИН-пад;
 - IP адрес - адрес сервера/интерфейса на котором принимаем соединения;
 - PORT - порт который слушаем;
 - IP адрес=0.0.0.0 - все адреса сервера;
- **server.cashregisters.binds** - тоже, что и server.pinpads.binds, только для касс;

Для версии Windows и Linux устанавливаются следующие параметры:

- **server.named.authTimeout=10000** - таймаут (в тысячных долях секунды. 10000=10 секунд) получения «Arcus ID» от кассы или ПИН-пад (параметр используется при работе специальной версии протокола Arcus2, оптимизированной для работы в многопользовательском режиме);
- **server.named.pinpads.binds=0.0.0.0:9303** - аналогично server.pinpads.binds, за исключением того, что перечисленные сервисы требуют наличия в протоколе «Arcus ID»;
- **server.named.cashregisters.binds=0.0.0.0:9304** - аналогично server.cashregisters.binds, за исключением того, что перечисленные сервисы требуют наличия в протоколе «Arcus ID».

4_1_2 Файл Pinpads.conf

Содержит настройки очередей ПИН-пад или пулов.

- **PPAD_1=192.168.10.112:9301, *:9303** – ПИН-пад с IP адресом 192.168.10.112 пришедший на порт 9301 будет добавлен в пул (очередь) с названием PPAD_1. Так же в очередь 'PPAD_1' будет добавлен любой ПИН-пад пришедший на порт 9303 (*:9303);
- **PPAD_2=*:9304, 192.168.10.114:*** - в очередь PPAD_2 будут добавлены:
 - *:9304 – любой ПИН-пад, пришедший на порт 9304;
 - 192.168.10.114:* - любой ПИН-пад, пришедший с IP 192.168.10.114 (маловероятная ситуация).
- ПИН-пад будет доставаться из очередей, по мере поступления запросов с ККМ в соответствии с server.pinpads.pooltype в server.conf;
- К какой кассе, какая очередь определяется файлом cashregisters.conf.

4_1_3 Файл Cashregisters.conf

Файл Cashregisters.conf содержит настройки очереди для касс.

- **192.168.10.212:9402=PPAD_1** - кассу, пришедшую с IP 192.168.10.212 на порт 9402, обслужит ПИН-пад из очереди PPAD_1;
- **192.168.10.213:*=PPAD_2** - кассу, пришедшую с IP 192.168.10.213, не важно, на какой порт, обслужит ПИН-пад из очереди PPAD_2;
- ***:9404=PPAD_2** - все кассы, пришедшие на порт 9404, обслужит ПИН-пад из очереди PPAD_2.



Если на момент соединения приложения ARCUS в очереди нет свободных устройств ПИН-пад, соединение между ККМ и сервером разрывается.



Если к серверу подключен всего один ПИН-пад и с ним соединяется подряд одна и та же касса, а соединение при этом поднимается достаточно быстро, то сервер может не заметить освободившийся ПИН-пад и принимать транзакции через одну. Для избегания подобной ситуации рекомендовано проводить транзакции через какой-то временной интервал. Описанная ситуация НЕ появилась на CommandLineTool.

4_1_4 Файл Log.conf

Сервис пишет логи в каталог logs.

- ArcusIPServer.bootstrap.[Дата].log - запуски, установки сервиса. Не настраивается.
- ArcusIPServer.log - настраивается через log.conf.



В конфигурации по умолчанию для простоты настройки разрешены все ip адреса для подключений:

- Порты для ПИН-пад 9301, 9302; ip – любые;
- Порты для ККМ-POS 9401, 9402; ip – любые.

При начальной настройке рекомендуется подключить один ПИН-пад по предустановленным настройкам, а затем усложнять схему подключения (использовать статические ip, четко обозначать порты, работать с очередями).

Подробнее о логировании в java:

- <http://logging.apache.org/log4j/1.2/manual.html>;
- <http://www.vipan.com/htdocs/log4jhelp.html>.

4_1_5 Пример настройки ARCUS NET SERVER v.1.3

Одна ККМ и один ПИН-пад

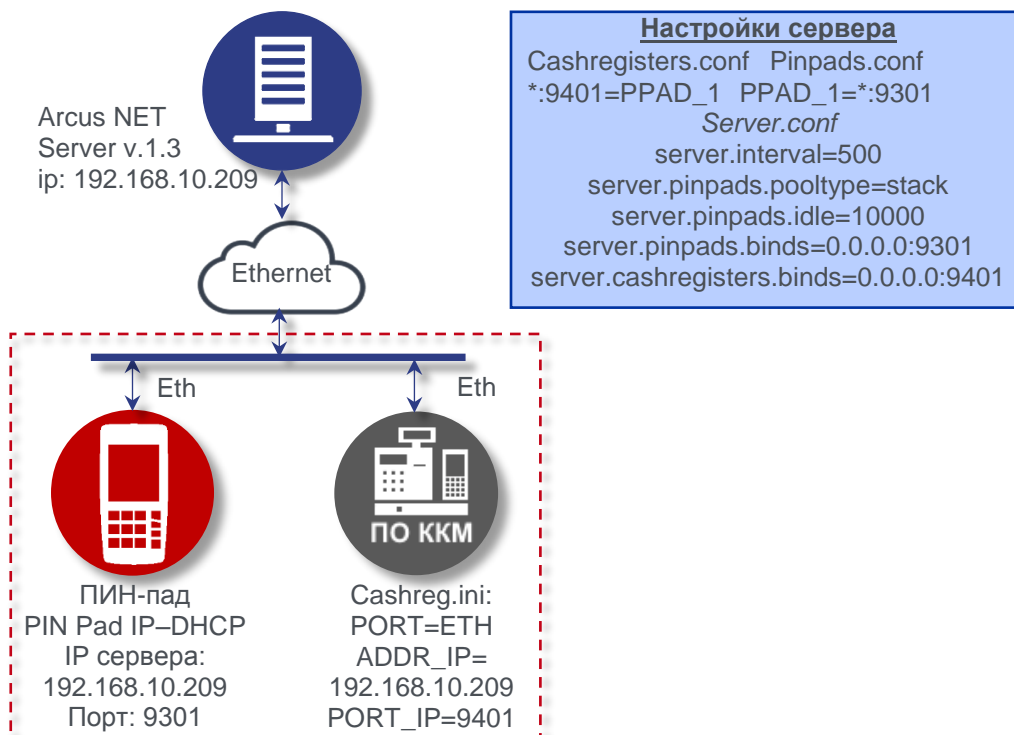


Рис. 9. Пример настройки сервера 1 ККМ-1 ПИН-пад

Несколько ККМ и несколько ПИН-пад

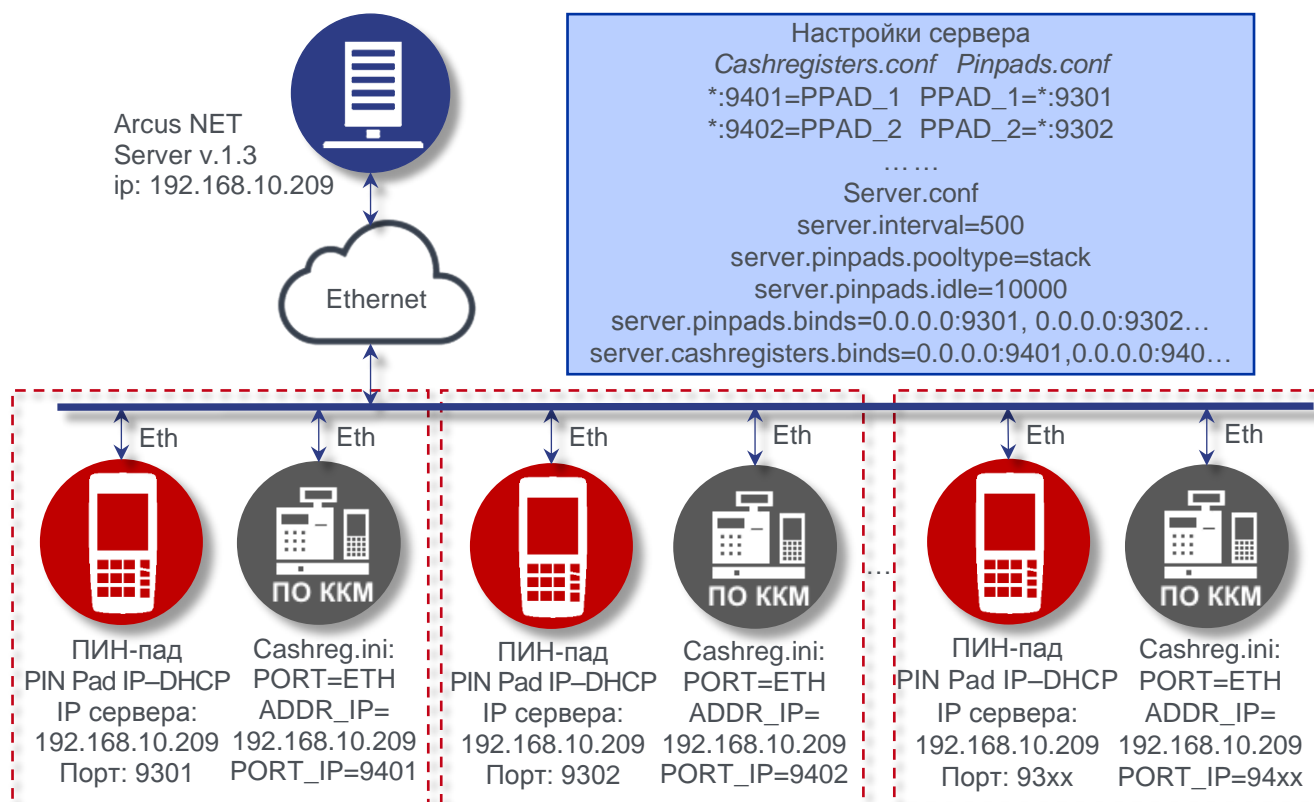


Рис. 10. Пример настройки сервера (более одной ККМ (ARCUS2) и более одного ПИН-пад)
- каждая касса работает со своим отдельным устройством ПИН-пад, это достигается за счёт
настройки нескольких очередей PPAD_1 и PPAD_2, ...

4_2 Arcus NET Server v.2

Arcus NET Server v.2 специально разработан для контрольно-кассовых машин (ККМ) с ограниченными ресурсами. Его основное отличие – передача параметров от кассы терминалу при помощи файлового обмена. Помогает сократить количество функций, выполняемых на ККМ.

Взаимодействие между платежными терминалами и ККМ может быть основано в двух вариантах, - файлового обмена или через TCP/IP.

4_2_1 Файловый режим

В файловом режиме для базовой версии приложение поддерживает выполнение следующих операций:

- оплата товара (выдача наличных);
- возврат товара (кредит, пополнение счета);
- отмена последней успешной операции;
- закрытие_дня (сверка итогов смены, пакета, дня — Z отчет);
- печать_журнала операций;
- меню_кассира (меню администратора).



Для процессингов Tieto и Openway реализованы операции ПРЕАВТОРИЗАЦИЯ и ЗАВЕРШЕНИЕ.

Система состоит из следующих компонент:

- Arcus NET Server – обеспечивает связь с терминалом, согласно спецификации протокола Arcus2. В качестве входных данных сервер принимает файлы с определенным именем, созданные на общем сетевом ресурсе⁸. Выходными данными являются файлы с определенными именами так же размещаемые на общем сетевом ресурсе или ответы кассе по TCP/IP.
- Терминал, работающий по протоколу Arcus2 и соединенный с Arcus NET Server по TCP.
- «ККМ модуль» – специальная внешняя компонента (программа или библиотека), формирующая файлы запросов на общем сетевом ресурсе и контролирующая получение и обработку файлов ответов (разрабатывается клиентом самостоятельно).

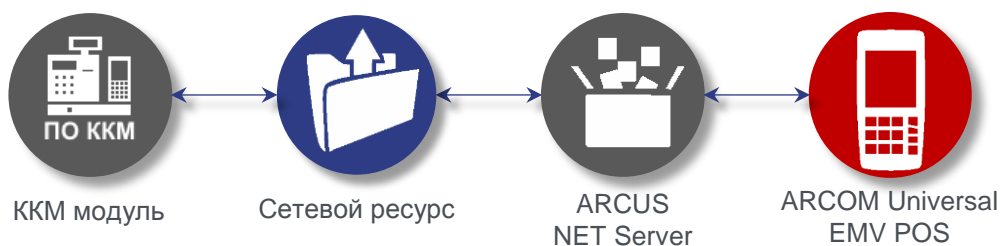


Рис. 11. Схема работы ARCUS NET Server v.2.0

Общие описание алгоритма работы системы:

1. **«ККМ модуль»** готовит файл с данными операции и записывает его в сетевую папку.
2. **Arcus NET Server:**

- определяет наличие нового файла в каталоге;
- производит анализ и чтение входных данных для операции;
- удаляет прочитанный файл;
- определяет устройство, на котором должна проводиться операция, в зависимости от номера кассы;
- проводит операцию на терминале в соответствии со спецификацией протокола ARCUS 2;

⁸ В данном случае общий сетевой ресурс – это локальный каталог на сервере, доступ к которому разрешен ККМ.

- подготавливает файл ответа, содержащий результат выполнения операции, и файл, содержащего образ слипа;
 - записывает эти файлы в тот же каталог, откуда был прочитан исходный файл.
3. **«ККМ модуль»** производит анализ и чтение файла с выходными данными, а также производит печать файла слипа. После чего файлы с данными и образом слипа удаляются.



В случае, если файл с выходными данными по истечении таймаута не забран кассой и не удален, то Arcus NET Server и проводит автоотмену операции и в случае успеха удаляет его, в противном случае файл остается в каталоге.

В приложении реализовано журналирование соединений и операций в отдельный файл.

4_2_2 Описание файлов

Файл запроса

ПО ККМ создает файл ХХННММSS.IN с данными, необходимыми для проведения операции и записывает его на общий сетевой ресурс.

ХХ – номер кассы от 01, до 99

ННММSS – время создания в формате - часы: минуты: секунды.

Файл может содержать следующие поля:

- operType = Код операции – обязательное поле для всех операций. Может принимать следующие значения:
 - 1 - ОПЛАТА
 - 2 - ОТМЕНА ПОСЛЕДНЕЙ УСПЕШНОЙ ОПЕРАЦИИ
 - 3 - ВОЗВРАТ
 - 4 - ОТМЕНА
 - 6 - ОТЧЕТ ПО ОПЕРАЦИЯМ
 - 7 - СВЕРКА ИТОГОВ
 - 98 - МЕНЮ КАССИРА
 - 99 - МЕНЮ АДМИНИСТРАТОРА ПИН-ПАДА
 - 100 - СЕССИЯ TMS
- amount = Сумма в копейках – обязательное поле для операций «оплата», «отмена», «возврат»;
- currency = Код валюты;
- track2 = Трек2 (для ручного ввода используется «M[PAN]=[EXP_Date]?») – опционально;
- originalAmount = Оригинальная сумма в копейках – для операций «отмена» и «возврат», заполняется, в том случае, если отменяемая или возвращаемая сумма отличается от исходной;
- terminalID = ID терминала – опционально;
- rgn = Ссылка – опционально;
- authCode = Код авторизации – опционально.



Если какое-то из обязательных полей отсутствует, для той или иной операции, то такая операция будет отклонена.

Пример файла запроса при параметре parser=shoko-file

TERM00.IN:

```
operType=1
amount=1200
currency=643
```

Пример файла запроса при параметре parser= properties-file

TERM00.IN:

```
operationType=1
```

Файлы ответа

После проведения операции на терминале Arcus Net Server формирует:

- ХХННММSS.SLP - файл с образом слипа в текстовом виде, сформированный в соответствии с настройками терминала.
- ХХННММSS.OUT – файл, содержащий результат выполнения операции, где:

ХХ – номер кассы от 01, до 99,

ННММSS – время создания в формате - часы: минуты: секунды.

Файл может содержать следующие поля:

- operType = Код операции;
- amount = Сумма в копейках;
- amountAuthorized = Авторизованная сумма - в общем случае может отличаться от суммы, поступившей в файле запроса;
- currency = Код валюты;
- PAN = Номер карты - маскированный номер карты, в соответствии с настройками ПИН-пада;
- Expiry = Срок действия карты (ГГММ);
- terminalID = ID терминала;
- rrn = Ссылка;
- authCode = Код авторизации;
- cardType = Название типа карты;
- date = Дата транзакции;
- time = Время транзакции;
- responseCode = Код ответа;
- textMessage = Расшифровка ответа;
- AID = EMV AID;
- applicationLabel = EMV ApplicationLabel;
- TVR = EMV TVR.

Пример файла ответа при параметре parser=shoko-file

TERM00.OUT:

```
operType=1
amount=1200
currency=643
AID=A0000000031010
approved=true
transactionDate=20140926
transactionTime=164731
authorizationCode=193709
PAN=*****6475
expiry=151031
applicationLabel=visa
authorizedAmount=1200
terminalId=20584938
rrn=426901246606
tvr=0000008000
cardholderName=IVAN IVANOV
cardTypeName=VISA
responseCode=0
textMessage=ОДОБРЕНО
```

Пример при параметре parser= properties-file:

TERM00.OUT

```
operationType=1
textMessage=Отказ ввода суммы
cardTypeName=ЕС/МС
terminalId=99999995
transactionTime=121131
uuid=a5e26735-8b48-491e-a14e-4d9a875e3b52
responseCode=203
approved=false
transactionStarted=false
```

```

authorizedAmount=0
expiry=160630
AID=A0000000041010
operationStarted=false
hostResponseCode=203
transactionDate=20150921
tvr=0000000000
applicationLabel=MasterCard
slip=
cardholderName=BOLDYREV/ALEXEY
PAN=*****3276
operation=PURCHASE
currencyCode=0

```

TERM00.SLP:

```

Терминал: 20584938
Чек: 87
Оплата
ОДОБРЕНО
СУММА: 12.00 RUR
AID: A0000000031010 Visa
TVR: 0000008000 TSI: F800
Карта: VISA PAN: *****6475
IVAN IVANOV
Номер ссылки RRN: 426901246606 EMV ICC
Код авторизации: 193709 Код ответа: 00
Дата (хост): 26/09/14 16:47:31
Дата (терминал): 26/09/14 16:47:21
ВВЕДЕН ОФФЛАЙН-ПИН

```

(Подпись кассира)



Сервис не работает с символическими ссылками.

4_2_3 Режим TCP/IP

Логика и принцип работы режима TCP/IP аналогичен режиму, который используется в ARCUS NET SERVER v.1.3.

4_2_4 properties-file Файлы настройки

Конфигурирование сервера осуществляется в файле services.xml. Этот файл содержит настройки для обоих режимов работы Arcus NET Server v.2.0.

Файл разделен на следующие разделы:


- <services* > - в разделе описаны запускаемые сервисы и их настройка:
 - *tcp-nio* - асинхронный tcp сервер. Может содержать тэги:
 - o Pinpads – сервис, обслуживающий терминалы;
 - o Cashregisters – сервис, обслуживающий ККМ – необходим только в случае работы в режиме с запросами по TCP.

/ Таблица 8. Атрибуты tcp-nio

Атрибут	Описание
bindAddr	Локальный IP адрес будет на котором будет запущен сервер. По умолчанию 0.0.0.0 - все адреса.
bindPort	Локальный порт, на котором будет запущен сервер. По умолчанию 9307
serverThreads	Количество серверных потоков (при большой нагрузке, чем больше, тем быстрее обрабатываются входящие соединения). По умолчанию 1.

Атрибут	Описание
clientThreads	Количество потоков, обрабатывающих соединения. 0- не ограничено, то есть пока не кончится память. По умолчанию 20. Т.е. после того, как пришел 20 клиент, каждый последующий клиент будут обслуживаться чуть медленнее. Чем больше это число, тем больше оперативной памяти требуется.
backlog	Максимальное количество клиентов, ожидающих соединения. По умолчанию 1024.
keepAlive	Включение SO_KEEPALIVE. Особого смысла на Windows не имеет, т.к. интервал на Windows - 2ч. По умолчанию false.
noDelay	Отключение TCP_NODELAY (Алгоритм Нейгла). По умолчанию false. При отключении алгоритма пакеты более фрагментированы - возрастает нагрузка на интерфейс, однако скорость отправки увеличивается.

/ Таблица 9. Атрибуты тэгов pinpads и cashregisters

Атрибут	Описание
idleTimeout	Таймаут, определяющий время бездействия терминала (мс). По умолчанию 5000.
inOperationIdleTimeout	Таймаут выполнения операции (мс). По умолчанию 120000.
maxActiveClients	Максимально количество подключенных устройств. При превышении ПИН-пад/ККМ не будут соединяться с сервером. По умолчанию 0 - не ограничено.
proto	Идентифицирует настройки, по которым работает ПИН-пад/ККМ (в текущей версии только openway).
encoding	Кодировка, в которой работает ПИН-пад/ККМ. По умолчанию "windows-1251". Выбранная кодировка должна поддерживаться установленной версией JRE/JDK.
authType	<p>Тип аутентификации:</p> <ul style="list-style-type: none"> • ip - идентификация клиента по адресу и порту (в этом случае указание name-mapping обязательно); • arcusid - идентификация клиента по Arcus IP ID. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;">  <p>Для работы в данном режиме необходимо при настройке «Работа с кассой» на терминале указать: «Use Arcus ID - да». Для идентификации терминала будет использоваться TMSID. Для работы с Arcus ID необходима специальная версия ArcCom.dll</p> </div>
name	Любая строка, которая будет выводиться в трассу.
Name-mapping	<p>Содержит настройки очередей ПИН-падов / ККМ. IP задается двумя способами:</p> <ul style="list-style-type: none"> • pattern – выражение; • mask – маска. <p>Параметры регистрочувствительные. bindName – название очереди, которая будет обрабатываться</p>
sslProfile	Атрибут для активации ssl шифрования – имя профиля ssl (содержится в поле name).

- fileService - сервис мониторинга новых файлов в указанной директории.
- fileTransactions - приложение обслуживающие файлы, как файлы транзакций на подключенных ПИН-падах.

/ Таблица 10. Атрибуты FileService

Атрибут	Описание
---------	----------

folder	Путь к общему сетевому ресурсу, в котором осуществляется мониторинг.
lockDetectionTimeout	Таймаут попытки открытия файла (мс). По умолчанию 2000.
lockDetectionDelay	Интервал между попытками открыть файл (мс). По умолчанию 500.

/ Таблица 11. Атрибуты тега fileTransactions

Атрибут	Описание
parser	Формат входных выходных файлов: <ul style="list-style-type: none"> • properties-file - базовый формат со всеми полями. • shoko-file - формат в рамках TID 788.
encoding	Кодировка входных/выходных файлов.
inputEnding	Расширение входных файлов. Регистрозависимый параметр.
printerEnding	Расширение файлов слипов. Регистрозависимый параметр.
outputEnding	Расширение выходных файлов. Регистрозависимый параметр.
cleanUp	Удаление входящих файлов после подбора – true/false
reversalTimeout	Таймаут удаления файла ответа (мс), если он останется после этого времени, то будет вызвана операция отмены. По умолчанию 30000. Должен быть больше 0 (НЕ отключается).
reversalLifetime	Время жизни реверсала (сек). Таймаут обнаружения освободившегося ПИН-пада и проведения реверсал. По истечении таймаута реверсал не будет проведен. По умолчанию 3600. 0 - бесконечно, до перезагрузки сервера. Если реверсал прошел успешно выходной файл и файл слипа удаляются.
Name-mapping	Имя файла задается одним из способов: <ul style="list-style-type: none"> • pattern – регулярное выражение (regex) • mask – маска (? – любой символ, * - любое количество любых символов) Параметры регистрозависимы. bindName – название очереди, которая будет обрабатываться



Параметры inputEnding, printerEnding, outputEnding и Name-mapping являются регистрозависимыми!

- `<*arcus-registry*>` - зарезервировано для будущего использования. Раздел должен содержать строку: `<arcus-registry type="stack"/>`
- `<loggers>` - в разделе содержатся настройки журналирования.

Все логи записываются в каталог размещения сервера Arcus NET Server в подкаталог logs/. Раз в сутки в каталоге logs/ создается подкаталог, с названием текущей даты и в него перемещаются и архивируются все логи.

Может содержать два подраздела:

- arcusConnectionsLogger - логирование соединений;
- FileLogger – записывает лог файл с немедленной записью на диск - рекомендовано для отладки, работает медленно;
- asyncFileLogger - асинхронно записывает лог файл блоками по 8196 байт - рекомендуется использовать на отлаженных продакшн системах. Работает быстрее, но при сбое, например, по питанию логи могут потеряться;
- arcusOperationsLogger - логирование операций;
- FileLogger;
- asyncFileLogger.

/ Таблица 12. Атрибуты тэгов FileLogger и asyncFileLogger

Атрибут	Описание
name	Атрибут, для идентификации конкретного логгера. Обязательный, но не обязательно уникальный. Значение по умолчанию: ArcusConnections
file	Имя файла, в который будет писаться лог. Обязательный, но не обязательно уникальный. Значение по умолчанию: connections.log
level	Устанавливает уровень информативности лога (RFU, на данный момент все сообщения пишутся уровнем INFO). Значение по умолчанию: DEBUG Необязательный атрибут,
maxDays	Максимальный срок жизни лога (в днях). По истечении этого времени лог, независимо от размера, будет заархивирован. Значение по умолчанию: 0-отключено
maxFiles	Максимальное число хранимых архивных файлов. Значение по умолчанию: 5
maxFileSize	Максимальный размер лога, при достижении этого размера лог будет заархивирован. Значение по умолчанию: 250MB
pattern	Формат сообщения в логе. Все возможные ключи описаны здесь, раздел Patterns. Значение по умолчанию: %d %p - %m%n
fileNamePattern	маска имени файлов архива. %d{} в фигурных скобках формат даты. %i - порядковый номер архивного файла. Значение по умолчанию: connections.log-%d{MM-dd-yyyy}-%i.gz



Количество упоминаний тех или иных тэгов ограничено только аппаратными возможностями ПК.

Для активации расширенного (debug) логирования укажите в файле настроек services.xml:

```
<configuration logConf="debug-logger.xml">.
```

Затем в файле "debug-logger.xml" можно выставить необходимый уровень логирования. Для использования новой настройки необходимо перезапустить службу.

4_2_5 Пример настройки ARCUS NET Server v.2.0

Пример реализации с файловым обменом

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
<services>
  <fileService name="File transactions" folder="D:/Temp/Exchange">
    <fileTransactions parser="shoko-file" encoding="CP866"
      printerEnding=".SLP" outputEnding=".OUT" inputEnding=".IN" cleanUp="true">
      <name-mapping pattern=".*" bindName="PPAD_1" />
    </fileTransactions>
  </fileService>
</services>
</configuration>
```

Пример реализации для 2 ККМ и 2 ПИН-пад в режиме ArcusID

Режим ArcusID необходим при работе кассового ПО в режиме удаленного рабочего стола (RDP-сессии), когда нет возможности определить ip-адрес кассы. В этом случае ArcusNETServer использует ArcusID и создаёт логическое соединение «ПИН-пад касса».

Для такой реализации используется **специальная версия ArcCom.dll** с поддержкой ArcusMulti, которая обеспечивает многократный запуск в режиме терминальной сессии.

ККМ должна передавать значение ArcusID. При настройке servise.xml ArcusID для сервиса ККМ и ПИН-ПАД задаётся в **authType="arcusid"**:

Для вызова arccom.dll с функцией multiuser:

```
int ProcessOwIdFull(char *ArcusIPID , UserAuthIntFull *auth_st, size_t size)
```

где ArcusIPID – любая NULL терминированная строка.

Если параметр ArcusIPID отличен от пустой строки или NULL, Arcus IP ID будет передаваться на Arcus IP Server. В остальном логика работы идентична *ProcessOwFull(UserAuthIntFull *auth_st, size_t size)*.



Функцию ProcessOwIdFull могут вызывать одновременно несколько приложений. Для корректной работы, параметр ArcusIPID должен быть идентичным для каждого приложения.



Библиотека использует одну и ту же конфигурацию для всех клиентов, поэтому изменение в работе с конфигурационными файлами не требуется, однако есть ряд файлов, которые нельзя использовать одновременно, так как они имеют одно и тоже название, что может повлечь трудности при работе в много пользовательском режиме. Поэтому в имена файлов будет автоматически добавляться ArcusIPID (если задано).

В случае, когда можно определить кассу по ip-адресу, допускается частичное использование режим ArcusID для подключения **терминалов** к ArcusNETServer. Это уменьшает размер файла servise.xml, упрощает настройку и повышает стабильность подключения. В этом случае в сервисе, обслуживающем ККМ, необходимо определить параметр «bindName», который будет использоваться в качестве значения «ArcusID», указанного при настройке терминала. При настройке servise.xml для сервиса терминалов определяется **authType="arcusid"**, а для **ККМ authType="ip"**. Пример такой конфигурации:

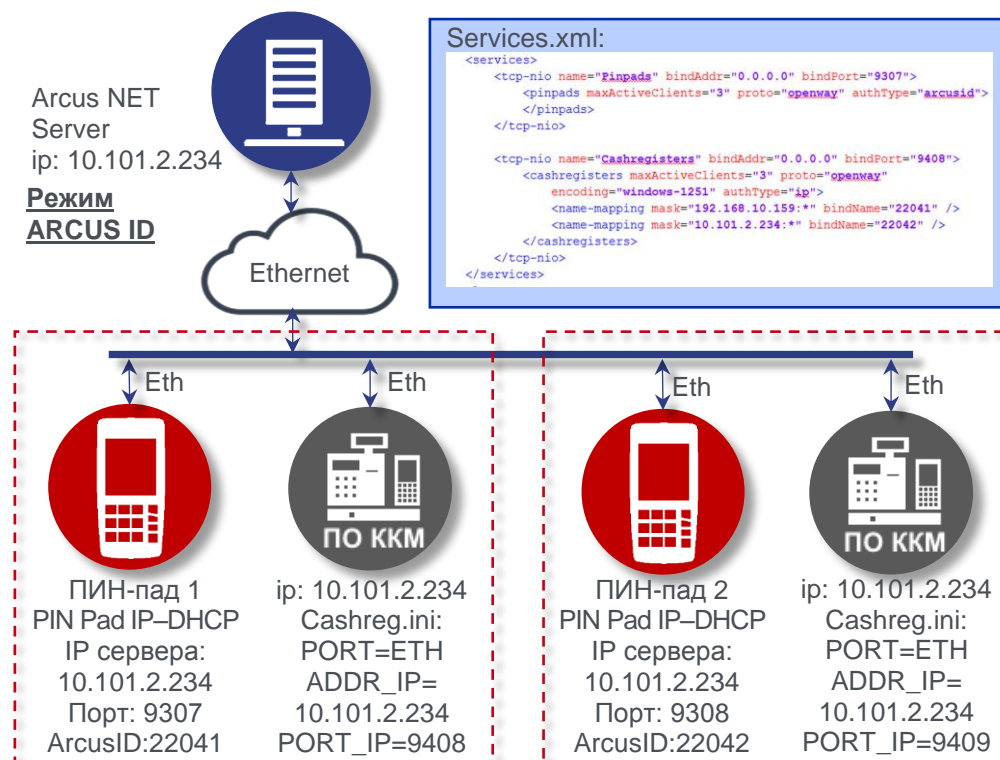


Рис. 12. Пример настройки Arcus NET Server v.2 – Режим Arcus ID

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <services>
    <tcp-nio name="pinpads" bindAddr="0.0.0.0" bindPort="9307">
      <pinpads maxActiveClients="3" proto="tiesto" authType="arcusid" name="IP_ID">
```

```

</pinpads>
</tcp-nio>
<tcp-nio name="Cashregisters" bindAddr="0.0.0.0" bindPort="9408">
  <cashregisters maxActiveClients="3" proto="tieto" encoding="windows-1251"
  authType="ip">
    <name-mapping mask="192.168.10.159:*" bindName="22041" />
    <name-mapping mask="10.101.2.234:*" bindName="22042" />
  </cashregisters>
</tcp-nio>
</services>
</configuration>

```

Пример реализации для 2 ККМ и 2 ПИН-пад в режиме IP

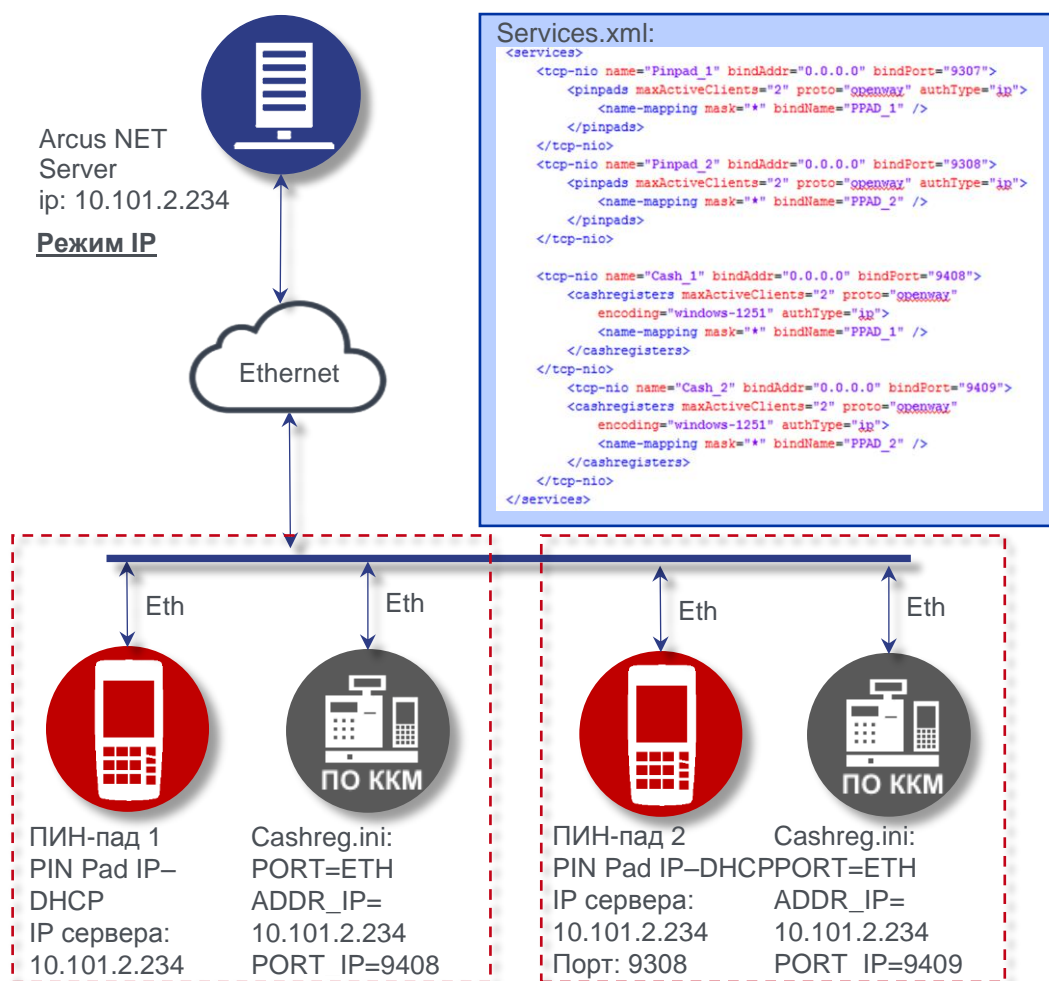


Рис. 13. Пример настройки Arcus NET Server v.2 – Режим IP

```

<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <services>
    <tcp-nio name="Pinpad_1" bindAddr="0.0.0.0" bindPort="9307">
      <pinpads maxActiveClients="2" proto="openway" authType="ip">
        <name-mapping mask="*" bindName="PPAD_1" />
      </pinpads>
    </tcp-nio>
    <tcp-nio name="Pinpad_2" bindAddr="0.0.0.0" bindPort="9308">
      <pinpads maxActiveClients="2" proto="openway" authType="ip">
        <name-mapping mask="*" bindName="PPAD_2" />
      </pinpads>
    </tcp-nio>
    <tcp-nio name="Cash_1" bindAddr="0.0.0.0" bindPort="9408">
      <cashregisters maxActiveClients="2" proto="openway"
        encoding="windows-1251" authType="ip">
        <name-mapping mask="*" bindName="PPAD_1" />
      </cashregisters>
    </tcp-nio>
  </services>
</configuration>

```



```

    </cashregisters>
  </tcp-nio>
  <tcp-nio name="Cash_2" bindAddr="0.0.0.0" bindPort="9409">
    <cashregisters maxActiveClients="2" proto="openway"
    encoding="windows-1251" authType="ip">
      <name-mapping mask="*" bindName="PPAD_2" />
    </cashregisters>
  </tcp-nio>
</services>
</configuration>

```

4_2_6 Пример файла лога

Пример файла настройки журналирования

```

<loggers>
  <arcusConnectionsLogger>
    <fileLogger name="ArcusConnections" file="connections.log"
    fileNamePattern="connections.log-%d{MM-dd-yyyy}-%i.gz" maxDays="0"
    maxFiles="5" maxFileSize="250 MB" pattern="%d %p - %m%n" level="DEBUG"/>
  </arcusConnectionsLogger>
  <arcusOperationsLogger>
    <fileLogger name="ArcusOperations" file="operations.log"
    fileNamePattern="operations.log-%d{MM-dd-yyyy}-%i.gz" maxDays="0"
    maxFiles="5" maxFileSize="250 MB" pattern="%d %p - %m%n" level="DEBUG"/>
  </arcusOperationsLogger>
</loggers>

```

Пример лога соединений

```

2014-10-24 19:33:56,314 INFO - 'PPAD_1:192.168.100.69:9307(1473)' registered as free
2014-10-24 19:33:58,730 INFO - 'PPAD_1:192.168.100.69:9307(1473)' was taken for work.
(busy)
2014-10-24 19:33:58,737 INFO - 'UCM:192.168.10.98:9308(49274) '-
>'PPAD_1:192.168.100.69:9307(1473)'started operation 'd2719058-616e-49be-9136
02b2ef30087e'
2014-10-24 19:34:01,295 INFO - 'PPAD_1:192.168.100.69:9307(1473)' registered as free
2014-10-24 19:56:19,252 INFO - 'PPAD_1:192.168.100.69:9307(1476)' registered as free
2014-10-24 20:02:16,231 INFO - 'PPAD_1:192.168.100.69:9307(1476)' was taken for work.
(busy)
2014-10-24 20:02:16,283 INFO - 'fileTransactions:D:\Temp\testExchange\SSS01P.IN'-
>'PPAD_1:192.168.100.69:9307(1476)' started operation 'b1ef5dc9-59ca-4c11-a07e-
13199b07fb35'
2014-10-24 20:02:30,018 INFO - 'PPAD_1:192.168.100.69:9307(1476)' registered as free

```

Пример лога операций

```

2014-10-24 19:53:45,668 INFO - START: 6cfb7eb9-c885-4160-9af3-aa065671251c, PURCHASE(1),
Amount:1001, Currency:643, PAN (last 4):**5015
2014-10-24 19:53:53,160 INFO - STOP: 6cfb7eb9-c885-4160-9af3-aa065671251c, PURCHASE(1),
Amount:1001, Currency:643, Card=VISA, PAN (last 4):**5015, DECLINED, RC:305,
HRC:305, Message:Карта не обслуживается
2014-10-24 21:20:59,791 INFO - START: 015b2792-80c7-4b5d-be8f-07121762b370,
PREAUTHORIZATION(8), Amount:1001, Currency:643
2014-10-24 21:21:10,783 INFO - STOP: 015b2792-80c7-4b5d-be8f-07121762b370,
PREAUTHORIZATION(8), TerminalId:20584939, Amount:1001, Currency:643, Card=VISA, hrsCrypt,
PAN (last 4):**5015, RRN=429701263745, DECLINED, RC:96, HRC:96, Message:Неисправность в системе

```

4_2_7 Пример настройки SSL

```

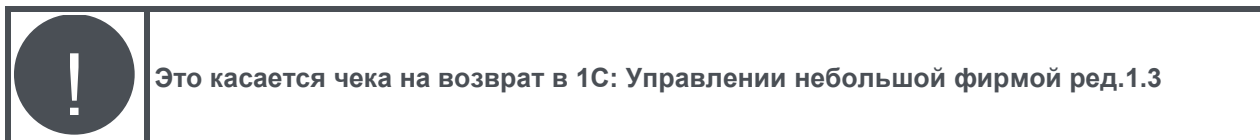
<sslProfiles>
  <sslProfile allTrust="true" auth="2" cacheTimeout="0" isClient="true" name="TEST">
    <protocols>TLSv1.2</protocols>
    <sslCertificateStore file="test.jks" keyPassword="test" password="testtest"
type="jks"/>
    <sslTrustedCertificatesStore file="trusted.jks" password="testtest" type="jks"/>
  </sslProfile>

```

4_2_1 Аварийная отмена покупки

Только для варианта поставки Б:

В программах 1С две различные операции «аварийная отмена покупки» и «возврат покупки» в некоторых случаях реализуются через один и тот же метод драйвера, что не позволяет дифференцировать их без дополнительного взаимодействия с пользователем:



В таком случае, при выполнении операции на экран будет выведен дополнительный диалог:

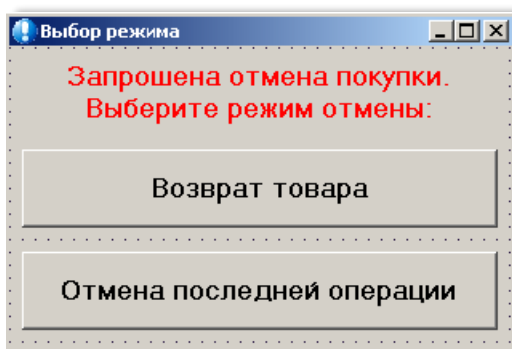


Рис. 14. Выбор режима

Пользователю необходимо выбрать, какую именно операцию он хочет провести, руководствуясь возникшей ситуацией.

5_Основные этапы установки и настройки ARCUS 2.1 CAP и ARCUS Net Server v.1.3

1. Установите Arcus NET Server v.1.3 из инсталляционного пакета (см. Руководство по установке ARCUS).
2. Измените настройки Arcus NET Server v.1.3:
 - В основном конфигурационном файле Server.conf;
 - В файле настройки очередей для ПИН-пад или пулов Pinpads.conf;
 - В файле настройки очереди для касс Cashregisters.conf;
 - В файле логов Log.conf.



Пример настройки Arcus NET Server v.1.3 для работы в режиме одна ККМ и один ПИН-пад рассмотрен в Приложении 1.
Пример настройки Arcus NET Server v.1.3 для работы в режиме более одной ККМ и более одного ПИН-пад рассмотрен в Приложении 2.

3. Установите ARCUS 2.1 CAP.
4. В файле cashreg.ini укажите:
 - Параметр **PORT** должен иметь значение **ETH**, настройка для работы при помощи Ethernet соединения: **PORT=ETH**;
 - Параметр **ADDR_IP=IP:PORT** IP адрес и порт компьютера, на котором установлен Arcus NET Server: **ADDR_IP=192.168.10.209:9401**.
5. Подключите терминал к сети при помощи Ethernet кабеля и подключите блок питания.
6. Переключите терминал в режим работы с кассой:
 - При перезагрузке или включении питания, дождитесь появления окна с сообщением «Для настройки работы с кассой нажмите 'F'» и нажмите кнопку 'F'. Следуйте диалогам настройки.



- **Ответы нажатием кнопок:**
- **зеленая = ДА**
- **красная = НЕТ**
- **выбор необходимого пункта из скролл-меню**

- Работа с кассой? -> ДА
- Выберите порт: ETH
- Введите IP адрес (на котором установлен Arcus NET Server).
- Порт, который указан в настройках сервера («pinpads.conf» и «server.conf»).
- «Use Arcus ID» -> НЕТ
- IP через кассу? -> НЕТ
- Диалог на кассе? -> НЕТ - все диалоги в процессе операции обслуживания карты будут вестись на устройстве ПИН-пад — экран и клавиатура; ДА - диалоги при проведении операции по карте будут выведены на экране ККМ и ответы — с клавиатуры кассы.
- Таймаут диалога? -> 12000 (в 10 x mS = 120 000 mS = 120 S = 2 min).
- Таймаут передачи данных? -> 1000 (в 10 x mS = 10 000 mS = 10 S).
- Принтер? -> На кассе (фискальный принтер).
- ШИРИНА ЧЕКА? -> 24 (до 42)



Настройка «Ширина чека» влияет только на ширину ЧЕКА. Для изменения ширины отчета необходимо задать шаблон отчета в TMS. По-умолчанию шаблон оптимизирован для ширины в 24 символа.

После окончания ввода параметров терминал произведёт попытку подключения к сервису и, в случае успеха, отобразит на экране стандартное сообщение «ГОТОВ К РАБОТЕ».

Если для связи с кассой используется Ethernet, и подключиться к серверу не удалось, терминал попытается произвести подключение повторно. Если за 4 попытки подключение не будет установлено, то терминал выведет на главный экран ожидания сообщение «НЕТ ПОДКЛЮЧЕНИЯ К КАССЕ».

5_1 Тестовая транзакция (Windows и Linux)

Проведите тестовую транзакцию, запустив **CommandLineTool**:
Формат запуска **CommandLineTool** следующий:

```
CommandLineTool /o[код_операции_ККМ] /c[код валюты] /a[Сумма]
```

где

- **Код операции ККМ** – выбранная кассиром операция;
- **Валюта** – код валюты операции;
- **Сумма** – сумма операции в минимальных единицах валюты, без разделителя.

Например:


```
C:\Arcus2\CommandLineTool\BIN> CommandLineTool /o1 /c643 /a12345
```

После выполнения команды будет произведена оплата на 123,45 руб.

ПРИЛОЖЕНИЕ 1

КЛЮЧИ ДЛЯ МЕТОДОВ SET/GET

Взаимодействие происходит через методы Set/Get(IN/OUT), в которые передается следующий набор ключей:

	Ключи регистрозависимые.
---	---------------------------------

Ключ	Назначение	Направление	Максимальная длина данных (рекомендованная ISO8583)
currency	Код валюты ISO	IN	3
amount	Сумма операции без разделителей (в минимальных единицах валюты)	IN	12
result_amount	Сумма после операции (в минимальных единицах валюты)	OUT	12
original_amount	Оригинальная сумма операции без разделителей (в минимальных единицах валюты)	IN	12
rfu	Трек2 карты (Track2 ISO)	IN	60
pan	Маскированный номер карты(PAN)	OUT	20
expiry	Expiry Date (YYMM)	OUT	4
aid	EMV AID	OUT	80
card_type	Тип карты	OUT	80
application_label	EMV ApplicationLabel	OUT	80
date	Дата (процессинг) (DDMM)	OUT	4
time	Время (процессинг) (hhmmss)	OUT	6
terminal_id	Идентификатор терминала TID (процессинг)	IN, OUT	8
ARCUS_ID	Идентификатор Аркус2 и АРМ кассира	IN	12
response_code	Код ответа- результат операции	OUT	3
auth_code	Код авторизации	IN, OUT	8
rrn	RRN (номер ссылки)	IN, OUT	12
tvr	EMV TVR	OUT	80
received_text_message	Текстовое	OUT	80
text_message	Расшифровка кода ответа	OUT	80
cardholder_name	Имя держателя карты	OUT	27
enc_data	Шифрованные данные карты для поддержки функционала HRS	IN,OUT	64
trace_id	ID-преавторизации	IN, OUT	60
original_date_time	Дата-время оригинальной операции (YYMMDDhhmmss)	IN, OUT	12

ПРИЛОЖЕНИЕ 2

ПРИМЕР ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ

Прототипы используемых функций

```
extern "C" __declspec(dllimport) void* __cdecl CreateITPos();
extern "C" __declspec(dllimport) void __cdecl DeleteITPos(void *obj);
extern "C" __declspec(dllimport) int __cdecl ITPosSet(void *obj, const char *key, const char *value, int size);
extern "C" __declspec(dllimport) int __cdecl ITPosGet(void *obj, const char *key, char *value, int size);
extern "C" __declspec(dllimport) int __cdecl ITPosRun(void *obj, int cmd);
```

- obj – объект взаимодействия с POS терминалом;
- key – название параметра, передаётся в виде ASCII строки (с терминальным символом);
- value – содержимое параметра (IN/OUT), передаётся в виде массива символов;
- size – длина данных, передаваемых в параметре, если -1, то длина вычисляется как для ASCII строки (только для нуль terminated строк);
- cmd - код операции, определенный в ops.ini.

Пример использования на языке C++



Под ОС Linux необходимо добавить дефайн `__LINUX__`

```
//-----
#include "stdafx.h"
#include <string>

#ifdef __LINUX__
    #include <stdexcept>
    #include <dlfcn.h>
#else
    #include <windows.h>
#endif
//-----
#ifdef __LINUX__
    #ifndef __cdecl
        #define __cdecl __attribute__((cdecl))
    #endif

    #define HINSTANCE void*
    #define LoadLibrary(x) dlopen((x), RTLD_LAZY)
    #define FreeLibrary(x) dlclose((x))
    #define GetProcAddress(x, y) dlsym((x), (y))
    #define GetLastError std::to_string(errno).c_str
#endif

typedef void *(__cdecl *ARCCOM_CREATE)();
typedef void(__cdecl *ARCCOM_DELETE)(void *);
typedef int(__cdecl *ARCCOM_SET)(void *, const char *, const char *, int);
typedef int(__cdecl *ARCCOM_GET)(void *, const char *, char *, int);
typedef int(__cdecl *ARCCOM_RUN)(void *, int);

#ifdef __LINUX__
    const char dll_name[] = "./libarccom.so";
#else
    const char dll_name[] = "Arccom.dll";
#endif
//-----
int main()
{
    void *pos_obj = NULL;
    ARCCOM_DELETE ArccomDelete = NULL;
```

```

try
{
    HINSTANCE dll = LoadLibrary(dll_name);

    if (dll == NULL)
    {
        throw std::runtime_error("DLL not Load");
    }

    ArccomDelete = (ARCCOM_DELETE) GetProcAddress(dll, "DeleteITPos");

    ARCCOM_CREATE ArccomCreate = (ARCCOM_CREATE) GetProcAddress(dll, "CreateITPos");
    ARCCOM_SET ArccomSet = (ARCCOM_SET) GetProcAddress(dll, "ITPosSet");
    ARCCOM_GET ArccomGet = (ARCCOM_GET) GetProcAddress(dll, "ITPosGet");
    ARCCOM_RUN ArccomRun = (ARCCOM_RUN) GetProcAddress(dll, "ITPosRun");

    if (ArccomCreate == NULL) throw std::runtime_error("function CreateITPos not loaded");
    if (ArccomDelete == NULL) throw std::runtime_error("function DeleteITPos not loaded");
    if (ArccomSet == NULL) throw std::runtime_error("function ITPosSet not loaded");
    if (ArccomGet == NULL) throw std::runtime_error("function ITPosGet not loaded");
    if (ArccomRun == NULL) throw std::runtime_error("function ITPosRun not loaded");

    pos_obj = ArccomCreate();

    if (pos_obj == NULL)
    {
        throw std::runtime_error("create object fail");
    }

    if (ArccomSet(pos_obj, "amount", "2000", -1) != 0) throw std::runtime_error("set amount fail");
    if (ArccomSet(pos_obj, "currency", "643", -1) != 0) throw std::runtime_error("set currency fail");

    int result = ArccomRun(pos_obj, 62);

    if (result != 0)
    {
        throw std::runtime_error("operation fail");
    }

    char rrn[100] = { 0 };

    if (ArccomGet(pos_obj, "rrn", rrn, sizeof(rrn) - 1) < 0)
    {
        throw std::runtime_error("get RRN fail");
    }

    printf("Command result: %d\n", result);
    printf("RRN: %s\n", rrn);
}
catch (std::exception &e)
{
    printf("ERROR: %s\n", e.what());
}
catch (...)
{
    printf("ERROR: unhandled exception\n");
}

if ((ArccomDelete != NULL) && (pos_obj != NULL))
{
    ArccomDelete(pos_obj);
}

return 0;
}
//-----

```

Пример использования на языке C#

```
using System;
using System.Runtime.InteropServices;
using System.Text;

namespace TestArccom
{
    class Program
    {
        [DllImport("Arccom.dll")]
        static public extern IntPtr CreateITPos();

        [DllImport("Arccom.dll")]
        static public extern IntPtr DeleteITPos(IntPtr pClassNameObject);

        [DllImport("Arccom.dll")]
        static public extern int ITPosSet(IntPtr pClassNameObject, byte[] key, byte[] value, int len);

        [DllImport("Arccom.dll")]
        static public extern int ITPosGet(IntPtr pClassNameObject, byte[] key, byte[] value, int len);

        [DllImport("Arccom.dll")]
        static public extern int ITPosRun(IntPtr pClassNameObject, int cmd);

        static void Main(string[] args)
        {
            IntPtr pos_obj = IntPtr.Zero;

            try
            {
                // create object
                pos_obj = CreateITPos();

                if (pos_obj == IntPtr.Zero)
                {
                    throw new Exception("create object fail");
                }

                // set require parameters
                ITPosSet(pos_obj, Encoding.ASCII.GetBytes("amount"), Encoding.ASCII.GetBytes("2000"), -1);
                ITPosSet(pos_obj, Encoding.ASCII.GetBytes("currency"), Encoding.ASCII.GetBytes("643"), -1);

                // start operation
                int res = ITPosRun(pos_obj, 62);

                byte[] rrn = new byte[20];

                // get result fields
                ITPosGet(pos_obj, Encoding.ASCII.GetBytes("rrn"), rrn, rrn.Length);

                Console.WriteLine("Result: " + res.ToString());
                Console.WriteLine("RRN: " + Encoding.ASCII.GetString(rrn));
            }
            catch (Exception e)
            {
                Console.WriteLine("ERROR: " + e.Message);
            }

            if (pos_obj != IntPtr.Zero)
            {
                // delete object
                DeleteITPos(pos_obj);
            }
        }
    }
}
```


Пример использования на языке Object Pascal (Delphi):

```
unit test;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls, dynlibs;

type
  { TArcusForm }

  TArcusForm = class(TForm)
    AmountEdit: TEdit;
    AdminButton: TButton;
    MessageEdit: TEdit;
    MessageLabel: TLabel;
    BalanceButton: TButton;
    ResultEdit: TEdit;
    ResultLabel: TLabel;
    PurchaseButton: TButton;
    CurrencyEdit: TEdit;
    AmountLabel: TLabel;
    CurrenyLabel: TLabel;
    procedure AdminButtonClick(Sender: TObject);
    procedure BalanceButtonClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure PurchaseButtonClick(Sender: TObject);
  private
    { private declarations }
    procedure StartArcus(cmd : integer);
  public
    { public declarations }
  end;

var
  ArcusForm: TArcusForm;

implementation

{$R *.lfm}

{ TArcusForm }

type
  CharArray = array[0..1024] of char;
  // функция создания объекта взаимодействия с терминалом
  // результат: указатель на объект для работы с ПОС терминалом, если "nil" - ошибка при создании объекта
  TCreateTPos = function () : Pointer; cdecl;
  // функция удаления объекта (после данной функции, использовать указатель нельзя, необходимо снова вызывать
CreateTPos)
  // входные данные
  // Ptr - указатель на объект
  TDeleteTPos = procedure (var Ptr : Pointer); cdecl;
  // установка параметра
  // входные данные
  // Ptr - указатель на объект
  // Key - название параметра, передаётся в виде ASCII строки с завершающим терминальным символом
  // Value - значение параметра, буффер данных, размером Size, если Size=-1, то длина вычисляется по терминальному
символу (для ASCII строк)
  // Size - размер передаваемых данных, -1 - размер вычисляется автоматически
  // результат: если 0 - параметр успешно записан, отличное от 0 - ошибка
  TTPosSet = function (Ptr : Pointer; Key : PChar; Value : PChar; Size : Integer) : Integer; cdecl;
  // получение параметра
  // входные данные
  // Ptr - указатель на объект
  // Key - название параметра, передаётся в виде ASCII строки с завершающим терминальным символом
  // выходные данные
  // Value - значение параметра, буффер данных
  // Size - размер зарезервированного буффера
  // результат: если 0 - параметр успешно записан, отличное от 0 - ошибка (наприме маленький размер буффера)
```

```

TITPosGet = function (Ptr : Pointer; Key : PChar; var Value : CharArray; Size : Integer) : Integer; cdecl;
// запуск операции на терминале
// входные данные
// Ptr - указатель на объект
// Cmd - номер команды, указанный в файле "ops.ini"
TITPosRun = function (Ptr : Pointer; Cmd : Integer) : Integer; cdecl;

var
// глобальный объект загруженной библиотеки
IHandle : TLibHandle;
// функция создания объекта взаимодействия с терминалом
CreateITPos : TCreateITPos;
// функция удаления объекта (после данной функции, использовать указатель нельзя, необходимо снова вызывать
CreateITPos)
DeleteITPos : TDeleteITPos;
// установка параметра
ITPosSet : TITPosSet;
// получение параметра
ITPosGet : TITPosGet;
// запуск операции на терминале
ITPosRun : TITPosRun;

// преобразываем буффер в строку (возможно не лучшее решение, в паскале не силён)
// есть проблема с преобразованием буффера с русскими буквами (win1251), получается строка из вопросов, видимо проблема
в кодировках
function ArrayToString(const a: array of char): string;
begin
if Length(a) > 0 then
SetString(Result, PChar(@a[0]), Length(a))
else
Result := "";
end;

// начало работы, обнуляем все указатели
procedure TArcusForm.FormCreate(Sender: TObject);
begin
IHandle := NILHandle;
CreateITPos := nil;
DeleteITPos := nil;
ITPosSet := nil;
ITPosGet := nil;
ITPosRun := nil;
end;

// процедура взаимодействия с терминалом
procedure TArcusForm.StartArcus(cmd : integer);
var
Res : Integer;
Buffer : CharArray;
PosPtr : Pointer;
begin
// если запускаем первый раз, то инициализируем библиотеку
if IHandle = NILHandle then
begin
IHandle := LoadLibrary('Arccom.dll');

if IHandle <> NILHandle then
begin
// инициализируем функции взаимодействия
Pointer(CreateITPos) := GetProcAddress(IHandle, 'CreateITPos');
Pointer(DeleteITPos) := GetProcAddress(IHandle, 'DeleteITPos');
Pointer(ITPosSet) := GetProcAddress(IHandle, 'ITPosSet');
Pointer(ITPosGet) := GetProcAddress(IHandle, 'ITPosGet');
Pointer(ITPosRun) := GetProcAddress(IHandle, 'ITPosRun');
end;
end;

// если какая то из функций не проинициализирована, выходим (лучше выходить с кодом ошибки)
if @CreateITPos = nil then exit;
if @DeleteITPos = nil then exit;
if @ITPosSet = nil then exit;
if @ITPosGet = nil then exit;
if @ITPosRun = nil then exit;

// создаём объект управления ПОС терминалом, данный объект передаём во всех остальных функциях
PosPtr := CreateITPos();

```

```

// если объект не создан, завершаемся (лучше выходить с кодом ошибки)
if PosPtr = nil then exit;

// используем try на всякий случай, хотя сама библиотека не генерирует исключений
try
  // указываем сумму операции
  ITPosSet(PosPtr, 'amount', PChar(AmountEdit.Text), -1);
  // указываем валюту
  ITPosSet(PosPtr, 'currency', PChar(CurrencyEdit.Text), -1);
  // запускаем операцию под номером "cmd"
  Res := ITPosRun(PosPtr, cmd);

  // в переменной Res получаем результат работы ПОС терминала: 0 - успех, в остальных случаях - ошибка
  ResultEdit.Text := IntToStr(Res);

  // чистим массив для получения дополнительной информации о прошедшей операции
  FillChar(Buffer[0], sizeof(Buffer), #0);

  // запрашиваем RRN операции
  ITPosGet(PosPtr, 'rrn', Buffer, sizeof(Buffer));

  MessageEdit.Text := ArrayToString(Buffer);
except
  showmessage('непредвиденная ошибка');
end;

// удаляем объект работы
DeletelTPos(PosPtr);
end;

// запускаем операцию ОПЛАТА
procedure TArcusForm.PurchaseButtonClick(Sender: TObject);
begin
  StartArcus(1);
end;

// запускаем операцию БАЛАНС (номер операции завит от настроек ops.ini)
procedure TArcusForm.BalanceButtonClick(Sender: TObject);
begin
  StartArcus(62);
end;

// запускаем меню администратора
procedure TArcusForm.AdminButtonClick(Sender: TObject);
begin
  StartArcus(99);
end;

end.

```

ПРИЛОЖЕНИЕ 3

МИНИМАЛЬНЫЙ ПЕРЕЧЕНЬ ПОДДЕРЖИВАЕМЫХ ОПЕРАЦИЙ

Интерфейс кассира кассового ПО должен поддерживать вызов модуля АРКУС2 для выполнения следующих финансовых и административных операций:

- **ТОРГОВЛЯ – (БАНК- ВЫДАЧА НАЛИЧНЫХ)**

- 1 ОПЛАТА ТОВАРА (ВЫДАЧА НАЛИЧНЫХ)
- 2 ВОЗВРАТ ТОВАРА (КРЕДИТ, ПОПОЛНЕНИЕ СЧЕТА)
- 3 ОТМЕНА ПОСЛЕДНЕЙ ПОКУПКИ (ОТМЕНА ВЫДАЧИ / КРЕДИТА)
- 4 ЗАКРЫТИЕ_ДНЯ (СВЕРКА ИТОГОВ СМЕНЫ, ПАКЕТА, ДНЯ — Z отчет)
- 5 ПЕЧАТЬ ЖУРНАЛА ОПЕРАЦИЙ
- 6 МЕНЮ_КАССИРА (МЕНЮ АДМИНИСТРАТОРА)

- **ОТЕЛИ, АЗС,РЕСТОРАНЫ**

- 7 ПРЕАВТОРИЗАЦИЯ (резервирование суммы) [опция]
- 8 ЗАВЕРШЕНИЕ РАСЧЕТА ПО ПРЕАВТОРИЗАЦИИ [опция]

- **ПРИ НАЛИЧИИ ТЕХ.ВОЗМОЖНОСТИ АРМ ТОРГОВЛИ**

- 9 ОТМЕНА ОПЕРАЦИИ ПО НОМЕРУ ЧЕКА [опция]

В большинстве случаев в ПО терминала определены 3 операции:

- ВОЗВРАТ ТОВАР - служит для возврата денежных средств на счёт клиента. В большинстве случаев выполняется если начальная операция оплаты была в другой смене (на другом терминале). Для проведения операции обязательно запрашивается карта клиента. Другие данные могут отличаться (номер чека, RRN, сумма) и настраиваются при конфигурации терминала. «Возврат» является кредитовой операцией с перечислением средств на счёт клиента, поэтому обрабатывается банком дольше (от 3х до 40 дней).
- ОТМЕНА ПОСЛЕДНЕЙ - это технологическая отмена, при которой не запрашиваются данные о проведенной транзакции (номер чека, номер RRN, PAN карты и д.р.). Команда вызывается кассовым ПО в случае, когда необходимо отменить последнюю успешную оплату (строго говоря, можно отменять и другие успешные транзакции, но необходимо уточнять такую возможность для каждого конкретного процессинга). В случае если транзакцию отменить нельзя, терминал выдаст сообщение "Невозможно отменить"
- УНИВЕРСАЛЬНАЯ ОТМЕНА - служит для отмены с обязательным указанием данных о начальной транзакции и предъявлением карты клиента. Запрашиваемые данные могут быть различны, и настраиваются при конфигурации терминала. В большинстве случаев Отмену можно проводить только если начальная операция оплаты была в этой же смене. В этом случае, чаще всего, для поиска начальной транзакции в журнале терминала используется RRN (хотя возможно использовать и номер чека). Операция «Отмена» обрабатывается на хосте достаточно быстро и средства поступают обратно на счёт клиента.



В связи с тем, что можно получить RRN, желательно его и использовать для операций «Универсальная отмена» и «Возврат».



Использование каждой из них необходимо обговаривать непосредственно с банком.

ПРИЛОЖЕНИЕ 4

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Список таблиц

/	Таблица 1. Хронология изменений.....	4
/	Таблица 2. Условные обозначения.....	4
/	Таблица 3. Глоссарий	4
/	Таблица 4. Аббревиатуры	6
/	Таблица 5. Описание файла настроек ARCUS.CFG ISO.....	10
/	Таблица 6. Описание файла настроек ARCUS.CFG UNIPAY	12
/	Таблица 7. Список операций для интеграции с ПО Arcom Universal EMV POS (NewWay)	19
/	Таблица 8. Атрибуты tcp-nio	35
/	Таблица 9. Атрибуты тэгов pinpads и cashregisters.....	36
/	Таблица 10. Атрибуты FileService	36
/	Таблица 11. Атрибуты тега fileTransactions	37
/	Таблица 12. Атрибуты тэгов FileLogger и asyncFileLogger	38

Список рисунков

Рис. 1. Схема работы ARCUS 2.1 CAP	8
Рис. 2. TLS шифрование в режиме IP через кассу	9
Рис. 3. Схема взаимодействия кассового ПО с модулем Arcus2.1 с использованием Arccom.DLL.....	15
Рис. 4. Банковский чек.....	16
Рис. 5. Чек с ошибкой	16
Рис. 6. Виртуальная клавиатура	23
Рис. 7. Общая схема взаимодействия	27
Рис. 8. Схема работы ARCUS NET Server v.1.3.....	28
Рис. 9. Пример настройки сервера 1 ККМ-1 ПИН-пад.....	31
Рис. 10. Пример настройки сервера (более одной ККМ (ARCUS2) и более одного ПИН-пад)	31
Рис. 11. Схема работы ARCUS NET Server v.2.0.....	32
Рис. 12. Пример настройки Arcus NET Server v.2 – Режим Arc ID	39
Рис. 13. Пример настройки Arcus NET Server v.2 – Режим IP	40
Рис. 14. Выбор режима	42