

ARCUS 2

для интеграции с кассовыми решениями

Руководство администратора

/ Версия 1.7

Содержание

Хронология изменений	5
Условные обозначения.....	6
Глоссарий	6
Аббревиатуры	8
Введение	9
1_ ARCUS 2 CAP	10
1_1 Состав программы, основные принципы ее функционирования.....	10
1_1_1 Использование TLS шифрования в режиме IP через кассу	11
1_2 Режим работы с кассой	11
1_2_1 Файл настроек ARCUS.CFG для Режимы работы с кассой терминального приложения ISO (Openway, SmartVista, Tieto, CTL)	12
1_2_2 Файл настроек ARCUS.CFG для Режимы работы с кассой терминального приложения Unipay (CompassPlus, SPDH)	14
1_3 Интеграция с ККМ	16
1_3_1 Формат и содержание файла данных output.dat	19
1_4 Вызов ARCUS2.....	20
1_4_1 Windows	20
1_4_2 DOS	21
1_4_3 Linux	23
1_5 Настройки ARCUS 2 CAP	24
1_5_1 Файл инициализации Cashreg.ini	24
1_5_2 Файл ops.ini для ISO8583.....	24
1_5_3 Чтение карты для ISO	26
1_5_4 Файл ops.ini для Compass+ и SPDH	26
1_5_5 Файл rc_conv.ini	27
1_5_6 Файл rc_res.ini	27
1_6 Эксплуатация	28
1_6_1 Интеграция ARCUS2 CAP с ПО ККМ	28
1_6_2 Виртуальная клавиатура	28
1_6_3 Способы и периодичность контроля правильности работы ПО	28
1_6_4 Аварийные ситуации и способы их устранения.....	29
1_6_5 Описание фрагментов файла лога	29
1_7 Специальные версии ARCUS2 CAP	29
1_7_1 ПО ККМ DKLINK с расчетом скидки на терминале.....	29
1_7_2 ПО ККМ ДАТАКПАТ с расчетом скидки на кассе	30
1_7_3 Возврат без предъявления карты.....	32

2_ ARCUS 2 HRS CAP	34
3_ Модуль AC1	35
3_1 Принцип работы механизма	35
3_2 Возможные особенности и ошибки	36
3_2_1 Проблемы при подключении	36
3_2_2 В 1С не реализованы сервисные функции	36
3_2_3 Ожидание	37
4_ Интеграция через RS232 или USB	37
4_1 Схема взаимодействия с ПО ККМ	37
5_ Низкоуровневая интеграция	38
6_ Arcus NET Server	39
6_1 Arcus NET Server v.1.3	39
6_1_1 Файл Server.conf	40
6_1_2 Файл Pinpads.conf	40
6_1_3 Файл Cashregisters.conf	41
6_1_4 Файл Log.conf	41
6_1_5 Пример настройки ARCUS NET SERVER v.1.3	42
6_2 Arcus NET Server v.2	44
6_2_1 Файловый режим	44
6_2_2 Описание файлов	45
6_2_3 Режим TCP/IP	47
6_2_4 properties-file Файлы настройки	48
6_2_5 Пример настройки ARCUS NET Server v.2.0	51
6_2_6 Пример файла лога	54
6_2_7 Пример настройки SSL	56
6_2_1 Аварийная отмена покупки	57
7_ Основные этапы установки и настройки ARCUS2 CAP и ARCUS Net Server v.1.3	58
7_1 Тестовая транзакция для ОС Windows	59
7_2 Тестовая транзакция для ОС DOS	59
7_3 Тестовая транзакция для ОС Linux	59
Приложение 1. Интеграция с использованием COM Object	60
Описание PCPOSTConnectorObj	60
Приложение 2. Настройка ПИН-пад	62
Приложение 3. Описание структуры	63

UserAuthint	63
UserAuthintFull	64

Приложение 4. Пример использования ArcCom.dll..... 66

Вызов UserAuthint	66
Вызов UserAuthintFull	67

Приложение 5. Минимальный перечень поддерживаемых операций 71

Приложение 6. Функционал DLS (опционально)..... 72

Режим легкой интеграции с кассой	72
Режим полной интеграции с кассой	72
Оплата	72
Оплата наличными с начислением бонусов и без	72
Возврат со списанием бонусов и без	72
Открытие смены	72
Инкассация	72

Приложение 7. Информация о документе..... 73

Список таблиц.....	73
Список рисунков	73

Хронология изменений



/ Таблица 1. Хронология изменений

Версия	Дата	Автор	Детали
1.7	05/05/2016	Власенко И.Л.	Команда 2.21 Инфотерминала – доступна для всех [#29900]
1.7	26/04/2016	Власенко И.Л.	Добавлен пример вызова UserAuthIntFull.
1.7	24/03/2016	Власенко И.Л.	Добавлено описание вызова специальной версии arcscm.dll с функцией Multiuser.
1.7	15/12/2015	Власенко И.Л.	Добавлено описание структуры UserAuthIntFull в приложение 3
1.6	21/09/2015	Власенко И.Л.	Добавлены примеры файлов при параметре parser=properties-file
1.6	03/09/2015	Власенко И.Л.	Добавлен пункт 1_2 Режим работы с кассой
1.5	10/06/2015	Власенко И.Л.	Добавлен раздел 1_7_3 Возврат без предъявления карты
1.4	29/04/2015	Власенко И.Л.	Добавлены изменения, связанные с печатью дубликата чека (для SV_05)
1.4	02/09/2015	Власенко И.Л.	Добавлено описание возвращаемых значений функции SetAmountUpdateCB в раздел 1_7_2 ПО ККМ ДАТАКРАТ с расчетом скидки на кассе.
1.4	15/01/2015	Власенко И.Л.	Добавлено описание параметров работы со сканером штрих кода терминала iSMP (только для CTL_03).
1.4	12/01/2015	Власенко И.Л.	Добавлено описание параметров файла лога Arcus2CAP.
1.4	01/12/2014	Власенко И.Л.	Пример настроек ARCUS NetServer v1.3 перенесен из Приложений в раздел 6_1_5 Пример настройки ARCUS NET SERVER v.1.3
1.4	20/11/2014	Власенко И.Л.	Добавлено Приложение 6. Функционал DLS (опционально)
1.4	17/11/2014	Власенко И.Л.	Обновлено описание ops.ini
1.3	10/10/2014	Власенко И.Л.	Добавлен п. 1_1_1 Использование TLS шифрования в режиме IP через кассу
1.3	18/09/2014	Власенко И.Л.	Изменен список обязательных параметров, передающихся с ККМ на терминал.
1.3	11/09/2014	Власенко И.Л.	Добавлен п. 1_7 Специальные версии ARCUS2 CAP
1.3	05.08.2014	Власенко И.Л.	Добавлен новый класс команд «8» для вызова пунктов динамического меню
1.3	28.07.2014	Власенко И.Л.	Добавлено Приложение 4. Пример использования ArcCom.dll
1.3	01.07.2014	Власенко И.Л.	Добавлен пункт 1_6_2 Виртуальная клавиатура
1.2	15.05.2014	Власенко И.Л.	Вынесение Описания протокола обмена в отдельный документ
1.1	22.11.2013	Власенко И.Л.	Добавлен раздел 6_ Arcus NET Server
1.0	01.02.2013	Чарков М.В.	Создание документа

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.
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.

Термин	Определение
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 провайдеpa 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, разработанное компанией ООО «Инженико», описывает этапы установки и настройки ARCUS2 – библиотеки, позволяющей проводить интеграцию Терминального платежного приложения Universal EMV POS¹ и кассового программного обеспечения. В документе описаны:

- ARCUS 2 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 HRS CAP - модификация программного приложения ARCUS 2 CAP, разработанная специально для торгово-сервисных предприятий, специфика работы которых связана с необходимостью проводить платежные операции без предъявления банковской карты.
- ARCUS 2 AC1 – модуль для интеграции систем автоматизации торговли на платформе 1С: 8.2 с терминалами Ingenico.
- 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.

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 CAP

Arcus 2 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 протокол обмена ARCUS2 и обеспечивая диалог кассир-терминал на экране кассы или терминала (Режим работы с кассой).

ARCUS 2 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.

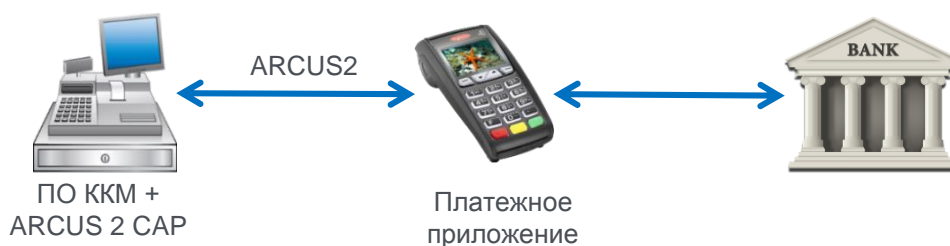


Рис. 1. Схема работы ARCUS 2 CAP

Программное решение работает на кассовом ПО в следующих операционных системах:

- Windows (XP SP3 и выше)
- MS DOS (версия 6.22)
- Linux (в некоторых случаях может потребоваться перекомпиляция модуля под конкретный тип операционной системы).
- Android 4.0.3 и выше (подробное описание предоставляется по запросу).

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 полностью³ 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 CAP является зависимой от поставщиков АРМ кассира частью. Сегодня на рынке кассового ПО существует широкий список поставщиков, отличающихся между

2 Pin-pad Application

3 Предыдущая версия ARCUS управление пин-падом брала на себя

4 ARCUS 2 CAP - Cash Register Application

собой как платформой (DOS, Windows/CE, Linux, UNIX, 1C, 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 – DOS, 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

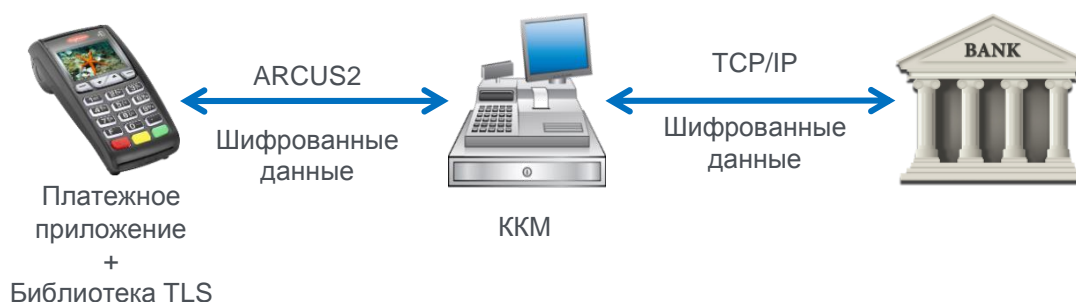


Рис. 2. TLS шифрование в режиме IP через кассу

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 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.

⁵ Вариант для Linux

3. Работа с кассой?

- НЕТ – терминал загружается в автономном режиме;
- 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 Режим 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	<ul style="list-style-type: none"> • 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		
<div></div> <div>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!</div>		
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
PARITY=NONE
STOP-BITS=1
ARCUSSERVER -IP=192.168.0.101
ARCUSSERVER -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 ТЕ_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

Параметр	Описание	Доступные значения
ARCUSSERVER -IP	IP для подключения к ARCUS Net Server	Значение в формате xxx.xxx.xxx.xxx
ARCUSSERVER -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		
<div><div></div><div>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!</div></div>		
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
ARCUSSERVER -IP=192.168.0.101
ARCUSSERVER -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. В зависимости от ОС ККМ:
 - а. **Windows:** Собранные кассовым ПО данные по операции записываются в структуру UserAuthInt / UserAuthIntFull.

- b. Затем ПО кассы вызывает функцию ProcessOW и в качестве параметра передает указатель на заполненную структуру UserAuthInt. Вызов функции ProcessOW является блокирующим. Управление диалогами с клиентом и кассиром передается приложению ПИН-пад.
 - c. **DOS, Linux:** Кассовое ПО осуществляет вызов исполняемого модуля ARCUS2 с передачей в качестве параметров перечисленных выше данных в указанном формате (см. п. 1_4 Вызов ARCUS2), после чего управление диалогами с клиентом и кассиром передается приложению ПИН-пад и модулю ARCUS2.
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. **Только для Windows:** Возвращаемым значением функции **ProcessOw** (), 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)

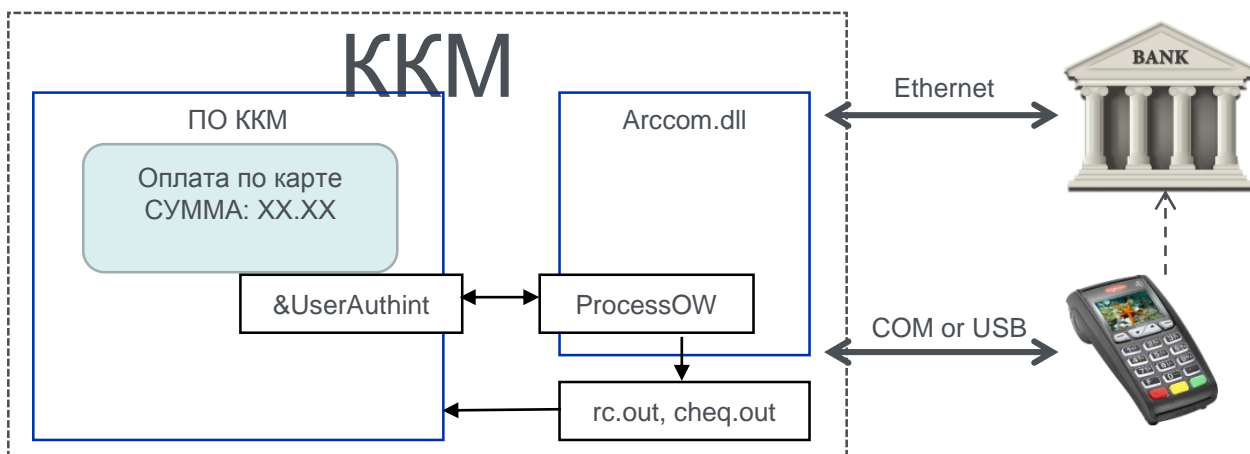


Рис. 3 Схема взаимодействия кассового ПО с модулем Arcus2 с использованием ArcCom.DLL

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

- Файл кода ответа (например, RESULT_FILE или rc.out), содержащий трехзначное значение кода 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 B
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
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, подключенного 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 из структуры **UserAuthInt**.

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 из структуры **UserAuthInt** 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 ARCUS2, которые не являются кодами ответа сервера Банка по безналичной операции.

1_4 Вызов ARCUS2


1_4_1 Windows

Стандартным способом интеграции с ККМ на базе MS Windows является использование библиотеки **Arcscm.dll** (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 ТМС).
- **/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[TracelD]** – ID предавторизации (доступно с января 2015г. только для продукта TE_02)
- **/h[Original Amount]** – общая авторизованная сумма (доступно с января 2015г. только для продукта TE_02)
- **/t[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).



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:

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

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

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»**.

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


- Передача данных трек 2, например, при чтении магнитной полосы на клавиатуре кассы. Оплата на 123.45 рублей, код валюты 643, трек карты «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?
```



Arcus2 исключен из области сертификации Payment Application Data Security Standards (PA DSS), так как при правильной настройке на хранит и не передает данные платежных карт в нешифрованном виде. Использование специальных версий программы CAP с возможностью считывания трека карты на ККМ, требует проведение сертификации ПО ККМ.



При использовании ARCUS HRS CAP для проверки шифрованных данных о карте, шифрованные данные передаются после ключа «/e».
Пример запуска:
CommandLineTool.exe /o1 /a2000
/e3AB5E25D6091053EAB71AFAE303B5327F3D713043819649F3D20D11F78A73811
Получение шифрованных данных настраивается в меню администратора - доп. настройки - исп. шифр. пар. (использовать шифрование параметров).
Данные шифруются с помощью 3DES CBC. Ключ для шифрования вводится в открытом виде в меню администратора – ПИН-пад - ARCUS ключ.

1_4_2 DOS

Формат запуска **cashreg.exe**:

cashreg.exe Код_операции_ККМ [Сумма] [Метод ввода карты] [Трек 2]

где

- **Код операции ККМ** – оплата, отмена, возврат и т.д. Значение кода соответствует коду операции в файле **ops.ini**.
- **Сумма** – сумма операции в минимальных единицах валюты, без разделителя.
- Метод ввода карты – 1 символ:
 - **'K'**, если номер карты и срок годности вводился вручную.
 - **'R'**, если карта считывалась на клавиатуре кассы.
- **Трек 2** – второй трек пластиковой карты. Если данные карты считывались через считыватель кассы (метод **«R»**), то данные 2-й дорожки предваряются символом **';** и заканчивается символом **'?'**.

Пример использования:

```
cashreg.exe 1 12345 R ;100001100013=0606101360000322?
```

Оплата на 123.45 рублей карта прочитана на кассе (ПО ККМ).

Если номер карты вводился вручную (метод **«K»**), то данные предваряются символом **«M»**. Номер карты и срок годности разделяются символом **«=»**. Срок годности передается в формате ММГГ.

```
cashreg.exe 1 12345 K M100001100013=0606?
```

Оплата на 123.45 рублей с ручным вводом номера карты на кассе.

Для возможности передачи номера карты из ПО ККМ в настройках ARCUS2 в файле cashreg.ini должна быть указана переменная **«NO_PA_DSS»**. В случае если переменная не определена, то данные о номере карты будут проигнорированы.



Обязательными параметрами для передачи являются: Код операции, валюту и сумму. Другие параметры могут отсутствовать. В случае необходимости терминал запросит недостающую информацию.

1_4_3 Linux

Параметры командной строки:

```
Cashreg /o[Код операции ККМ] /c[Код валюты] /a[Сумма] /i[Идентификатор терминала] /v[Код авторизации] /r[Ссылка] /e[Данные карты] /t[Трек 2]
```

где:

- **/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 разделителя.
- **/i[Идентификатор терминала]** - ID терминала до 9 символов (определяется ТМС).
- **/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.

В большинстве случаев обязательными параметрами необходимо передавать: Код операции, валюту и сумму. Другие параметры могут отсутствовать.



Для запуска cashreg в ОС Linux ему необходимо назначить права на исполнение. Например: \$ cashreg +x cashreg

Передача данных трек 2, например, при чтении магнитной полосы на клавиатуре кассы:

```
./cashreg /o1 /c643 /a12345 /t;1000023000000034=1106?
```

- оплата на 123.45 рублей, код валюты 643. Данные трек 2 предваряются символом «;» и завершаются «?».

В случае эмуляции ручного ввода данные передаются следующим образом:

M[Номер карты]=[срок действия в формате ГГММ]?

```
./cashreg /o1 /c643 /a12345 /tm1000023000000034=1106?
```

Для возможности передачи номера карты из ПО ККМ в настройках ARCUS2 в файле **cashreg.ini** должна быть указана переменная «NO_PA_DSS». В случае если переменная не определена, то данные о номере карты будут проигнорированы.



В большинстве случаев обязательными параметрами необходимо передавать: Код операции, валюту и сумму. Другие параметры могут отсутствовать. В случае необходимости терминал запросит недостающую информацию



Arcus2 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. Использование специальных версий программы CAP с возможностью считывания трека карты на ККМ, требует проведение сертификации ПО ККМ.

1_5 Настройки ARCUS 2 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).



С февраля 2016 года для касс на базе DOS, использующих сетевой драйвер PC-TCP, работающих с ПО для SV, в случае, если на терминал приходит команда, не прописанная в ops.ini, на терминал ничего не передается, а в лог ARCUS записывается строка «Illegal Command».

/ Настройки операции:

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

Op code - код операции для кассы.

Op class - класс операции (1 - финансовые, 2 - технические, 3 - административные).

Op id - ID операции.



В версии ARCUS2, выпущенной после июля 2014 года введен дополнительный класс команд 8 – команды динамического меню.

Для класса 1:

- 1 - OP_PURCHASE_SMS – Оплата
- 2 - OP_PURCHASE_DUAL
- 3 - OP_PREAUTHORIZATION – Предавторизация
- 4 - OP_CONFIRMATION – Подтверждение
- 5 - OP_UNIVERSAL_REVERSAL – Отмена (OP_CANCEL)
- 6 - OP_UNIVERSAL_REVERSAL_ADVICE – Отмена (OP_CANCEL_ADVICE)
- 7 - OP_BALANCE_INQUIRY – запрос баланса
- 8 - OP_MINI_STATEMENT – мини выписка
- 11 - OP_REFUND – Возврат
- 12 - OP_CASH_SMS - Выдача наличных
- 13 - OP_CASH_DUAL
- 14 - OP_CREDIT
- 15 - OP_CREDIT_VOUCHER
- 16 - OP_UTILITY_PAYMENT
- 17 - OP_BILLING_PAYMENT_SMS
- 18 - OP_BILLING_PAYMENT_DUAL

Для класса 2:

- 0 - Печать журнала.
- 1 - Сверка итогов.
- 2 - Меню кассира.
- 3 - Технический реверсал последней операции.

Для класса 3:

- 1,2 - Синхронизация времени.
- 4 - Сессия TMS.



Для класса 8 Op id должен соответствовать значению, указанному в TMS – Терминальные группы – Динамическое меню – Касса (может принимать значения от 1 до 20).

1=1,1,ОПЛАТА ТОВАРА
3=1,11,ВОЗВРАТ ТОВАРА

Автоматизированные Банковские системы (АБС)

1=1,12,ВЫДАЧА НАЛ.
3=1,14,КРЕДИТ
5=1,7,ЗАПРОС БАЛАНСА

Отели и заправочные станции

5=1,3,ПРЕАВТОРИЗАЦИЯ
8=1,4,РАСЧЕТ
9=1,15,CREDIT_VOUCHER

Отмены

2=2,3,ОТМЕНА ПОСЛЕДНЕЙ

4=1,5,УНИВЕРСАЛЬНАЯ ОТМЕНА

Отчеты и закрытие смены

6=2,0,ОТЧЕТ
16=2,10,КРАТКИЙ ОТЧЕТ
7=2,1,СВЕРКА ИТОГОВ

Специальные функции (совместимы со специальными версиями ПО пин-пада)

10=2,4,ЧТЕНИЕ КАРТЫ
11=9,2,ЧТЕНИЕ КАРТЫ2
12=9,7,ЧТЕНИЕ КАРТЫ3
12=9,3,ЧТЕНИЕ КАРТЫ4

60=2,20,ЗАПРОС ШТРИХКОДА – поддержка сканера штрих-кода терминала iSMP. Только для продукта с кодом CTL_03. Подробнее см. Universal EMV POS, руководство администратора. (ArcomDII от 8.12.2014 и выше).

Специальные функции –тест связи с сервером авторизации

51=2,6, ТЕСТ СОЕДИНЕНИЯ С ХОСТОМ - используется для проверки связи с банком, но при этом проверяется только возможность соединения с ним, без авторизации на хосте, отправки 0,138,ПЛАТЕЖ ему команды и печати чека.

55=9,8, ТЕСТ СВЯЗИ ТИЕТО - команда теста связи с хостом (начиная с версии, выпущенной после января 2015 г. данная команда удалена).

56=2,18 ТЕСТ СВЯЗИ SmartVista - команда теста связи с хостом - авторизация, отправка echo запроса, печать чека по результатам (начиная с версии, выпущенной после января 2015 г. данная команда выполняет так же отправку сообщения на хост).

Меню и административные операции

13=2,2,МЕНЮ КАССИРА
99=3,5,СЛУЖЕБНОЕ МЕНЮ
100=3,4,СЕССИЯ ТМС

Дополнительные функции

80=3,1,СИНХРОНИЗАЦИЯ ВРЕМЕНИ

2,21,ИНФО ТЕРМИНАЛА

214=2,14,печать дубликата чека (только для продукта SV_05)

- Для первой строки: если в командной строке прислан код операции 1, то будет выполнена первая операция первого класса (OP PURCHASE SMS), на экране будет отображено название «ОПЛАТА».



Как правило, изменения в данном файле требуются только в том случае, если поставщик кассового программного обеспечения в процессе интеграции использовал другие коды операций.

16=2,19, ОТЛОЖЕННАЯ ОПЕРАЦИЯ (SV).

1_5_3 Чтение карты для ISO

В терминальном ПО определены несколько команд для чтения карт:

- 2,4 – чтение магнитной карты (track1 и 2)
- 9,2 – аналогично 2,4. Добавлено для совместимости со старой SmartVista
- 9,3 – аналогично 2,4, но снижены требования к формату треков
- 9,7 – чтение магнитной или чиповой карты

Терминал возвращает значение номера карты клиента в маскированном виде. В большинстве случаев возвращаются первых 6 и/или последние 4 цифры, однако возможны особенности в зависимости от банка.

Если при интеграции необходимо работать с полным номером карты (или SHA1) необходимо использовать специализированную сборку ПО, которая не маскирует данные – предоставляется по запросу. В этом случае кассовой компании необходимо иметь сертификат Payment Application Data Security Standards (PA DSS).

В случае необходимости работы с картой рекомендовано использовать HRS версию платежного софта и ArcCom.dll, которые шифруют данные карты по 3DES алгоритму (поддерживается только в Windows).

1_5_4 Файл ops.ini для Compass+ и SPDH

1=0,128,(SPDH) ОПЛАТА ПОКУПКИ
3=0,130,(SPDH) ВОЗВРАТ
8=0,133,РЕЗЕРВИРОВАНИЕ
9=0,134,ПОДТВЕРЖДЕНИЕ РЕЗЕРВА
10=0,224,(SPDH) ОТМЕНА ПОСЛЕДНЕЙ
11=0,161,ЗАКРЫТИЕ СМЕНЫ (SHIFT)
12=0,162,ЗАКРЫТИЕ ДНЯ (DAY)
13=0,187,ПЕЧАТЬ ЖУРНАЛА
21=0,144,КОРРЕКЦИЯ ПОКУПКИ
22=0,225,(SPDH) ОТМЕНА ПО ЧЕКУ
23=0,147,КОРРЕКЦИЯ ПОПОЛНЕНИЯ НАЛИЧНЫМИ
24=0,148,КОРРЕКЦИЯ ВЫДАЧИ
25=0,136,ЗАПРОС БАЛАНСА
26=0,132,ВЫДАЧА НАЛИЧНЫХ
28=0,142,ПЕРЕВОД С КАРТЫ НА КАРТУ
29=0,143,ПОПОЛНЕНИЕ КАРТЫ НАЛИЧНЫМИ
30=0,176,КОПИЯ ПОСЛЕДНЕГО ЧЕКА
31=0,177,КОПИЯ ЧЕКА ПО НОМЕРУ
32=0,179,ЛОКАЛЬНЫЕ ИТОГИ
33=0,182,ДЕТАЛИ EMV
34=0,186,ОБЩИЙ ЖУРНАЛ
96=0,166,ТЕСТ СВЯЗИ
97=0,167,(SPDH) ЗАГРУЗКА ПАРАМЕТРОВ
98=0,50,МЕНЮ КАССИРА
99=0,53,АДМИНИСТРАТИВНОЕ МЕНЮ
100=0,204,TMS
120=0,160,ЗАКРЫТИЕ ПАКЕТА (BATCH)

Специальные команды

0,138,ПЛАТЕЖ – команда для отправки платежа в сторону третьих лиц (только для продукта ТРТР_04).

1_5_5 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_6 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 содержанию файлов инициализации находится в имеющемся 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 ARCUS2 CAP с ПО ККМ

ARCUS 2 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

Для торговых точек с ПО ККМ на базе Windows, при взаимодействии с процессингами WAY4 Card Management System, CTL, SmartVista (BPC), Tieto, Cortex, TITP, с июля 2014 года доступна модифицированная версия библиотеки, в состав которой входит поддержка виртуальной клавиатуры для ввода данных в режиме работы с кассой.



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 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 CAP и выполнению тестовых испытаний на стенде кассовой компании или в ТСП.

В случае выхода обновлений существующей версии ARCUS 2 CAP или выпуске принципиально новой версии ARCUS 2 CAP, осуществляются мероприятия по контролю совместимости интерфейса и настроек модуля ARCUS 2 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 CAP, краха ПО или ОС) техническими специалистами ТСП и службы тех. обслуживания эквайринга, осуществляются мероприятия по восстановлению и настройке модуля ARCUS 2 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 CAP, составлению отчетов по выявленным причинам и передача информации в службы поддержки программных продуктов и департамент разработки ПО ARCUS 2 CAP для проведения работ по устранению ошибок ПО ARCUS 2 CAP и выпуску обновления, после чего происходит выполнение отладочных, тестовых и пусковых испытаний в службах тестирования производителя ARCUS 2 CAP и в ТСП.

1_6_5 Описание фрагментов файла лога

- USE_JL ON\OFF - соответствует включенному или выключенному параметру USEJL в cashreg.ini. Параметр сигнализирует об использовании нескольких юридических лиц на терминале. Если параметр включен – в лог передается значение, указанное в usejl.txt (порядковый номер виртуального терминала или TerminalID):

USEJL ON, value=00010145

- Версия сборки ArcCom.dll
-

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

1_7 Специальные версии ARCUS2 CAP

Существуют реализации ARCUS 2 CAP для кассового ПО различных производителей, поддерживающие индивидуальную логику работы.

Реализация может осуществляться путем внесения изменений в библиотеку ARCUS2, добавлением дополнительных файлов настройки и пр. В некоторых случаях необходимо использовать модифицированные версии платежного приложения Universal EMV POS с поддержкой дополнительных функций.

1_7_1 ПО ККМ DKLINK с расчетом скидки на терминале

Данная модификация предназначена для продукта с кодом SPDH_03.

При оплате банковской картой с BIN, принадлежащим определенному диапазону, начисляется скидка. Если BIN карты не попадает в диапазон, оплата проводится обычным способом.

Дополнительные настройки оплаты Аркус-2 производятся в файле arcus.cfg.

- Скидки при оплате Аркус-2: 0 – выключены (простая оплата – по умолчанию), 1 – включены (оплата со скидкой банка).

- Код валюты: по умолчанию – 643 (рубли).
- Признак магазина, в котором установлено оборудование: 0 – супермаркет (по умолчанию, 1 – гипермаркет).

В чеке может быть только одна оплата ARCUS2 со скидками банка независимо от состояния параметра «Разрешить смешанную оплату», т.е. если в чеке есть наличная оплата, то нельзя добавить оплату Аркус-2 и наоборот, если в чеке есть оплата Аркус-2, то нельзя добавить другой тип оплаты. Чек должен быть полностью оплачен типом оплаты ARCUS2.

При оплате со скидкой банка помимо обычных параметров кассовый модуль (КМ) дополнительно передает два параметра:

- /dmin[сумма скидки в копейках] – минимальная сумма скидки по всему чеку.
- /dmax[сумма скидки в копейках] – при работе в гипермаркете максимальная сумма скидки; /dmax[сумма товаров на которые распространяется скидка] – при работе в супермаркете сумма товаров, на которые распространяется скидка.
- /a - сумма чека в копейках без скидок.

/ Логика расчета скидки

1. Расчет суммы авторизации:

Для гипермаркета

Сумма, переданная в параметре /a, умножается на размер скидки по банковской карте, после чего определяется допустимость предоставления получившейся скидки исходя из параметров /dmin и /dmax. Если рассчитанный дисконт получился менее /dmin, то предоставляется скидка в размере /dmin, если рассчитанный дисконт получился больше /dmax, то предоставляется дисконт в размере /dmax, во всех других случаях предоставляется дисконт, определенный терминалом.

Для супермаркета

Из суммы, переданной в параметре /a вычитается /dmax, умноженная на размер скидки по карте Банка. Если рассчитанный дисконт получился менее /dmin, то предоставляется скидка в размере /dmin, если рассчитанный дисконт получился больше /dmax, то предоставляется дисконт в размере /dmax, во всех других случаях предоставляется дисконт, определенный терминалом.

Пример 1:

В чеке обычный товар на который возможна скидка в диапазоне от 0 до 8%, клиент владеет дисконтной картой 5%:

/a13100 /dmin655 /dmax1048

Пример 2:

В чеке с единственным товаром стоимостью 131.00 руб., на который установлено ограничение по скидке 8% (т.е. максимально допустимая скидка на данный товар – 8%), магазин предоставляет скидку 5%:

/a 13100 /dmin655 /dmax1048

/dmin655 = (131.00 * 5%) * 100

/dmax1048= (131.00 * 8%) * 100

Пример 3:

В чеке только акционный товар, т.е. ограничение по скидке – 0%, не допускается скидка на товар:

/a13100 /dmin0 /dmax0

- Кассовое ПО, после получения суммы авторизации, выполняет обратный расчет и регистрирует процент выполненной скидки в журналах работы кассы.

1_7_2 ПО ККМ ДАТАКРАТ с расчетом скидки на кассе

Данная модификация предназначена для продукта с кодом OW_cless.

/ Логика расчета скидки

1. При вызове Безналичной оплаты на ККМ, кассовое ПО передает на модуль Universal EMV POS/ARCUS2 набор данных с указанием типа операции, кода валюты и суммы.
2. Кассир считывает карту на терминале, который в ходе обработки данных, если необходимо, запрашивает дополнительные действия, например, ввод ПИН.
3. Терминал формирует запрос итоговой суммы операции, в котором передает на ККМ первые 6 цифр PAN карты без спецсимволов.
4. ККМ, получив запрос, взаимодействует с дисконтным модулем кассовой системы с целью расчета итоговой суммы операции. Полученную сумму кассир подтверждает на кассе.
5. Если сумма подтверждена, то она передается терминалу, в противном случае операция сбрасывается и на терминал посылается отмена проведения транзакции.
6. После получения новой суммы, терминал формирует транзакционный пакет и отправляет запрос на банковский хост.
7. Полученный от банка ответ, обрабатывается терминалом и на ККМ передаются результаты авторизации, а также образ чека.

В случае если ККМ не получила ответ от терминала в течение заданного промежутка времени, а также если нет возможности распечатать чек о совершенной операции, кассовое ПО должно инициировать выполнение операции «Технический реверсал». Операция «Технический реверсал» предназначена для отмены последней операции, отправленной на выполнение на терминал.

/ Изменения в интерфейсе ARCUS2:

Для передачи кассе первых шести цифр номера карты и передачи новой суммы терминалу, в интерфейс библиотеки «arcscm.dll» добавляется следующий вызов:

```
/*!  
 * \brief describes Callback-function for calculation of net amount.  
 *  
 * \param [in] pan - pointer to buffer witch contains first 6 digits of PAN (null-terminated  
string)  
 * \param [out] newamount- pointer to buffer for storing new amount (must be null-terminated  
string, maximum length of string is 12 chars)  
 * \return - 1 if transaction must be continued  
 * \return - 0 if transaction must be terminated  
 * */  
typedef int (__cdecl *cbUpdateAmount)(const char pan[7], char newamount[13]);  
  
/*!  
 * \brief Sets callback-function for calculation of new amount.  
 *  
 * \param [in] func - pointer to callback-function.  
 * */  
extern "C" __declspec(dllexport) void __cdecl SetAmountUpdateCB(cbUpdateAmount func);
```

Функция позволяет передать указатель на функцию, которая непосредственно взаимодействует с дисконтным модулем кассы, и рассчитывает новую сумму операции.

Начиная с версии библиотеки, выпущенной после февраля 2015 года, для контроля вызова функции SetAmountUpdateCB добавлены следующие возвращаемые значения:

```
/*!  
 * \brief Sets callback-function for calculation of new amount.  
 *  
 * \param [in] func - pointer to callback-function.  
 * \return 0 - callback was setted  
 * \return 1 - callback was NOT setted  
 * */  
extern "C" __declspec(dllexport) int __cdecl
```

```
SetAmountUpdateCB(cbUpdateAmount func);
```

В остальном интерфейс библиотеки не меняется.

Вызов функции «SetAmountUpdateCB» необходимо выполнять каждый раз перед вызовом функции «ProcessOW». В процессе выполнения транзакции, после получения первых шести цифр номера карты, модуль Universal EMV POS/ARCUS2 вызовет функцию, указатель на которую был передан в качестве параметра функции «SetAmountUpdateCB». При этом 6 цифр номера карты будут переданы в параметре «pan». Функция должна вернуть код ответа 1, и новую сумму в параметре «newamount» для продолжения операции, или код ответа 0 для прерывания операции.



Функционал расчета скидки и итоговой суммы кассовым приложениям не применим к операциям по бесконтактным картам.

1_7_3 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.

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 журналу 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 "UserAuthInt.authCode" без каких либо преобразований.

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

- Для интеграции через CommandLineTool (XXXX - последние 4 цифры номера карты) /tDXXXX

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

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

Пример команды Arcus2:

```
CommandLineTool.exe /o6 /a22222 /tD6416 /v669993_B
```

2_ARCUS 2 HRS CAP

Программный продукт Arcus 2 HRS CAP (Hotel Reservation Service Cash register Application) – это модификация программного приложения ARCUS 2 CAP, полностью поддерживающая все его функции. Arcus 2 HRS CAP разработан специально для торгово-сервисных предприятий, специфика работы которых связана с необходимостью проводить платежные операции без предъявления банковской карты.

Основные различия:

При проведении безналичного платежа в версии Arcus 2 HRS CAP кассовое ПО в дополнительном поле может передавать криптограмму данных карты:

- номер карты;
- срок действия карты.

Шифрование данных осуществляется с использованием предварительно загруженного ключа.

Для реализации дополнительных функций в платежное приложение Universal EMV POS для терминалов Ingenico на платформах Telium 1,2 и Unicapt32, включены два дополнительных пункта:

- «ПИНПАД» – «Ключ ARCUS».
- «Включение передачи зашифрованных данных» (точное наименование может меняться, в зависимости от версии ПО).



Рис. 7. Схема работы ARCUS 2 HRS CAP

При выполнении рекомендаций по настройке, в соответствии с «Руководством по внедрению стандарта PA DSS», ARCUS 2 HRS CAP соответствует требованиям стандарта безопасности PA DSS.

Несмотря на то, что на кассу данные карты передаются в зашифрованном виде, ПО ККМ также должно быть сертифицировано на соответствие стандарту PA DSS.

Комплект библиотек ARCUS 2 HRS CAP реализован в виде DLL файлов и предназначен для использования только в ОС семейства Windows.

Настройка и эксплуатация ARCUS 2 HRS CAP аналогична Arcus 2 CAP.

3_Модуль AC1

Для интеграции систем автоматизации торговли на платформе 1С: 8.2 с терминалами Ingenico разработан модуль «Обработка Эквайринга ARCOM-Arcus2» (далее AC1).

Используя компоненту AC1 Кассовое ПО 1С взаимодействует с Arcus 2 CAP.

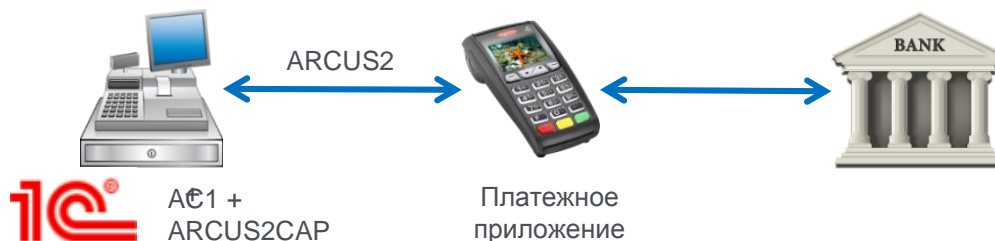


Рис. 8. Схема работы Модуля AC1

AC1 представляет собой внешнюю обработку обслуживания торгового оборудования, разработанную в соответствии с требованиями фирмы 1С, не использует никаких внешних компонент стороннего производителя.

В комплект поставки входит ряд настроечных файлов для ПО Arcus2CAP, настраивающий его на работу с 1С.

Подключение и настройка торгового оборудования производится из приложения 1С. Компонента AC1 передается пользователю в исходных кодах и при необходимости может быть адаптирована под требования заказчика специалистами 1С.



В данном документе описана настройка Модуля AC1 для Технологической платформы 8.2.15.294 и выше. В зависимости от редакций ПО 1С установка и настройка могут отличаться. В следующем разделе описана установка для следующих редакций:

Вариант поставки А:

1С: Управление торговлей ред. 10.3 (10.3.13.2) и выше

1С: Розница ред.1.0. (1.0.15.6) и выше

Вариант поставки Б:

1С: Управление торговлей ред. 11 (11.0.7.13) и выше

1С: Розница ред.2.0. (2.0.3.17) и выше

1С: Управление небольшой фирмой ред.1.3. (1.3.2.5) и выше.



Для других редакций и продуктов 1С может потребоваться доработка на стороне 1с. Для уточнения деталей обратитесь в нашу службу поддержки: support@ingenico.ru.

3_1 Принцип работы механизма

Вариант поставки А:

Механизм представляет собой внешнюю обработку обслуживания торгового оборудования, разработанную в соответствии с требованиями фирмы 1С. Версия API 2.0.5. Механизм не использует никаких внешних компонент стороннего производителя. Только объекты, поставляемые ООО «Инженико». Для связи с ПИН-пад используется COM-объект «ArcCom.PCPOSTConnectorObj.1» расположенный в «ArcCom.dll». В составе пакета поставляются файлы настроек для ARCUS2, которые используются для работы с 1С.

Вариант поставки Б:

Механизм представляет собой внешнюю компоненту, разработанную по стандартам технологии COM. В составе пакета поставляются файлы настроек для ARCUS2, которые используются для работы с 1С и другие файлы.

3_2 Возможные особенности и ошибки

3_2_1 Проблемы при подключении

Если после завершения основного этапа установки вместо сообщения «Готово» на экране появилось сообщение об ошибке, причины могут быть следующие:

Для вариантов поставки А и Б:

ПО «Инженико» не установлено или установлено неверно – не зарегистрировано в системном реестре. Выполните следующие действия:

- Находясь в каталоге установки ARCUS2 перейдите в каталог «DLL».
- Выполните команду `regsvr32 ArcCom.dll`.
- Должно появиться сообщение об успешной регистрации.

Для варианта поставки Б:

- ПО ARKOM-1C Connector не установлено или установлено неверно:
- Находясь в папке установки «ARKOM-1C Connector» выполните команду:
`regsvr32 SBRFCOMObject.dll`
- Проанализируйте ошибку, если она появится. Иногда для работы компоненты может не хватать прав пользователя Windows или обновлений операционной системы.

3_2_2 В 1С не реализованы сервисные функции

В программах 1С штатно не реализованы две важные сервисные функции «**загрузка параметров**» и «**отмена последней транзакции**».

Для варианта поставки А:

Эти функции разработчики механизма реализовали в форме параметров.

- Сервис – торговое оборудование - подключение и настройка торгового оборудования.
- Закладка «Эквайринговые системы»
- Выберите «ИНЖЕНИКО»
- Нажмите «Параметры»
- Нажмите «Функции»

Для варианта поставки Б:

Эти функции разработчики механизма реализовали в дополнительной обработке.

- Меню Администрирование – раздел «Сервис» - «Глобальные дополнительные обработки»:
- «Сервисные функции эквайринга ARKOM»
-

Настройки функций для обоих вариантов поставки:

Загрузка параметров загружает в пин-пад параметры процессингового центра.

Отмена последней операции отменяет последнюю покупку, сделанную по карте.



**Происходит отмена только факта оплаты в банке.
Деньги возвращаются на карту клиента.
Документ продажи в 1С, если он существует, не отменяется.**

При успешном выполнении операции будут распечатаны слип-чеки.

3_2_3 Ожидание

Если на экране зеленое окно с сообщением «УЖЕ ОТМЕНА» и идет время, значит эквайринговая система ждет нажатия кнопки Enter от пользователя.

- Если Enter не будет нажат до истечения времени, система вернет ошибку «нет кода ответа».
- Это штатное функционирование системы.

4_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.

4_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.

Собранные кассовым ПО данные по операции записываются им в структуру UserAuthInt. Затем ПО кассы должно вызвать библиотечную функцию ProcessOW, в качестве параметра передав ей указатель на заполненную структуру UserAuthInt. Вызов функции ProcessOW 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.

5_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.

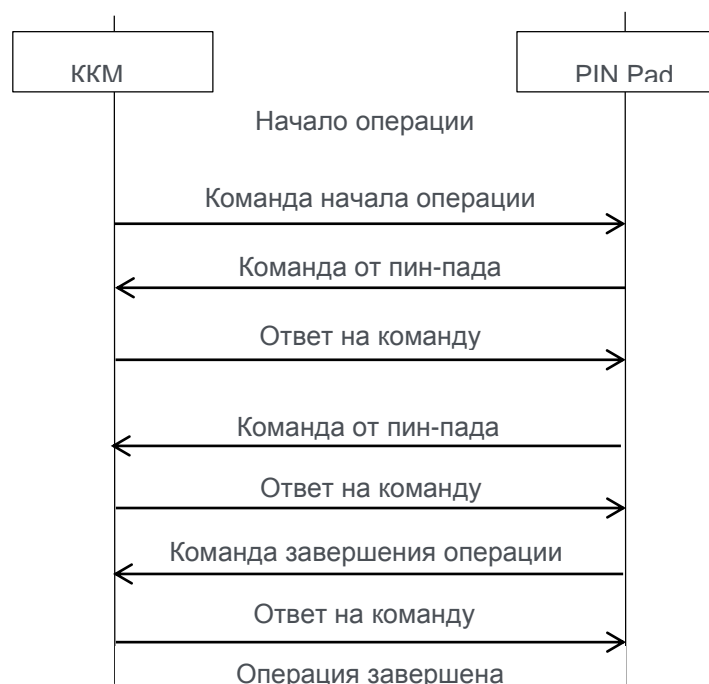


Рис.9. Общая схема взаимодействия

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

6_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 логге 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.

6_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. 10. Схема работы ARCUS NET Server v.1.3

При использовании Arcus NET Server существует несколько схем работы:

- Одна ККМ, один ПИН-пад;
- 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.

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

- 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

6_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».

6_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`.

6_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**.

6_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>

6_1_5 Пример настройки ARCUS NET SERVER v.1.3

Одна ККМ и один ПИН-пад

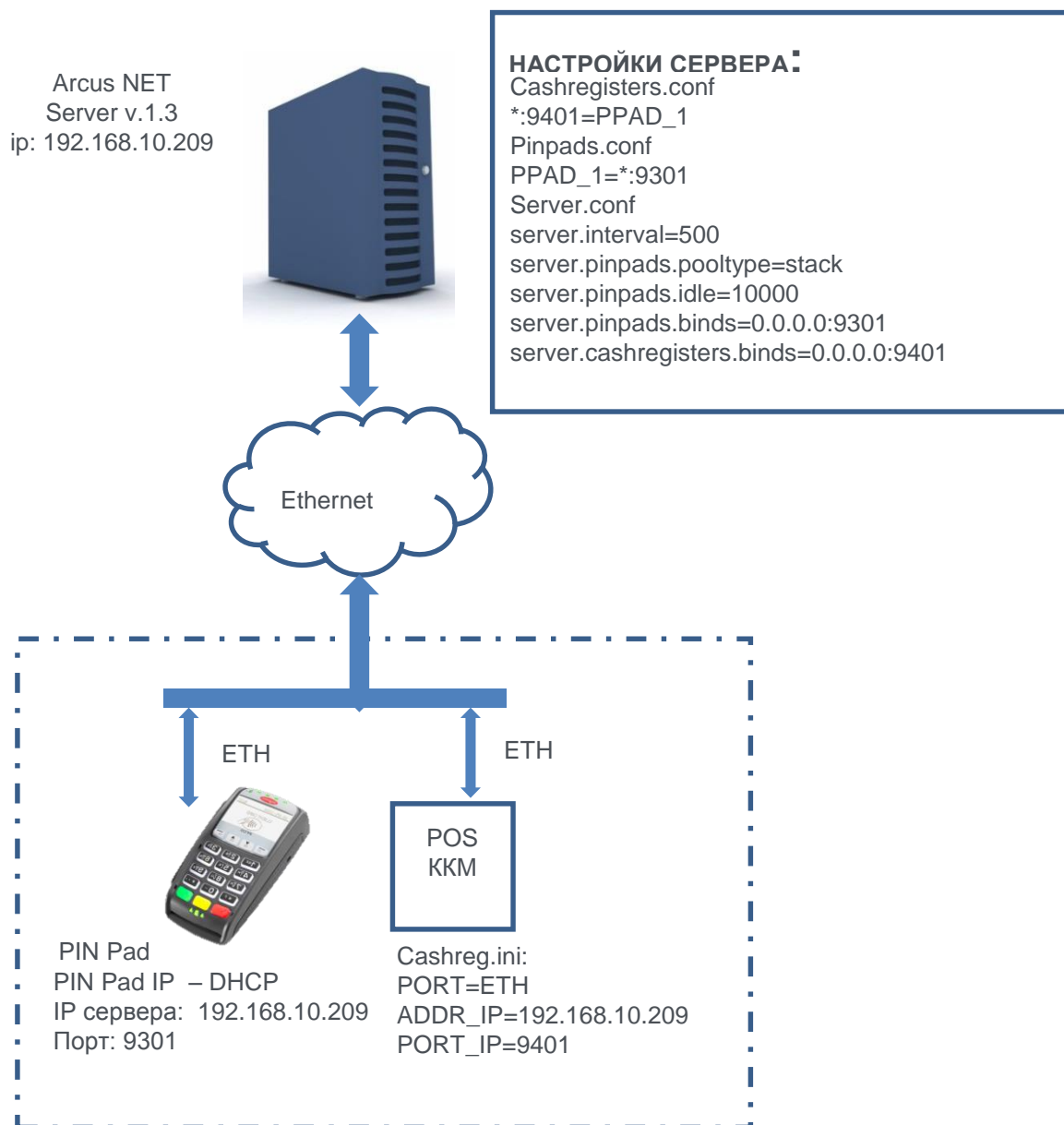


Рис. 11. Пример настройки сервера 1 ККМ-1 ПИН-пад

Несколько ККМ и несколько ПИН-пад

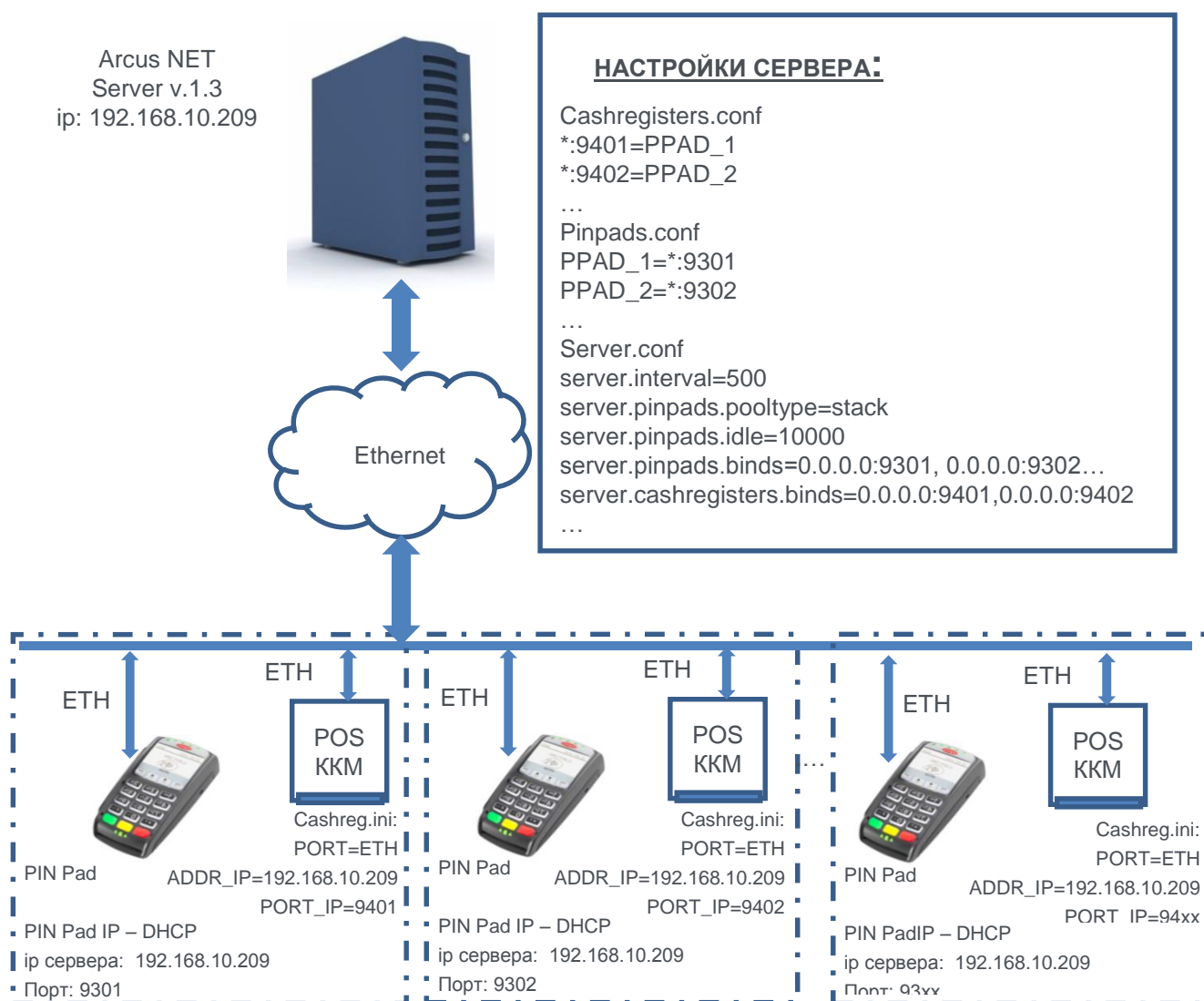


Рис. 12. Пример настройки сервера (более одной ККМ (ARCUS2) и более одного ПИН-пад)

Каждая касса работает со своим отдельным устройством ПИН-пад, это достигается за счёт настройки нескольких очередей PPAD_1 и PPAD_2, ...

6_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, - файловый обмен или через TCP/IP.

6_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 нескольких 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).

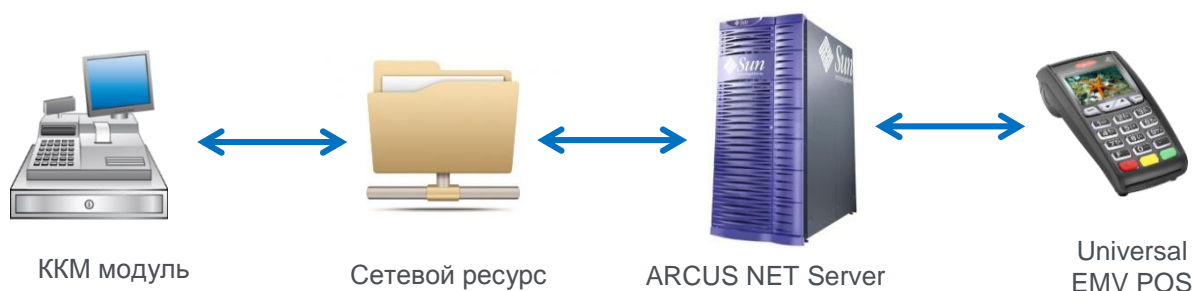


Рис. 13. Схема работы ARCUS NET Server v.2.0

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.

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

- 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.
 - 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 таймута 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.

6_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:

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

XX – номер кассы от 01, до 99,

HHMMSS – 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=000008000
cardholderName=IVAN IVANOV

cardTypeName=VISA
responseCode=0
textMessage=ОДОБРЕНО

Пример при параметре parser= properties-file:

TERM00.OUT

operationType=1
textMessage=Отказ ввода суммы
cardTypeName=EC/MC
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.

6_2_3 Режим TCP/IP

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

6_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:
 - Pinpads – сервис, обслуживающий терминалы
 - Cashregisters – сервис, обслуживающий ККМ – необходим только в случае работы в режиме с запросами по TCP.

/ Таблица 7. 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.
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 8. Атрибуты тэгов pinpads и cashregisters

Атрибут	Описание
idleTimeout	Таймаут, определяющий время бездействия терминала (мс). По умолчанию 5000.
inOpertionIdleTimeout	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

Атрибут	Описание
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>  <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 / KKM. 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. Должен содержать теги:
 - fileTransactions - приложение обслуживающие файлы, как файлы транзакций на подключенных пин-падах

/ Таблица 9. 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 10. Атрибуты тега fileTransactions

Атрибут	Описание
parser	<p>Формат входных выходных файлов:</p> <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	<p>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).</p>

Атрибут	Описание
reversalLifetime	Время жизни реверсала (сек). Таймаут обнаружения освободившегося пинпада и проведения реверсала. 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: 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 11. 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

Атрибут	Описание
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.

6_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 service.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 то же название, что может повлечь трудности при работе в многопользовательском режиме. Поэтому в имена файлов будет автоматически добавляться ArcusIPID (если задано).

В случае, когда можно определить кассу по IP-адресу, допускается частичное использование режима ArcusID для подключения **терминалов** к ArcusNETServer. Это уменьшает размер файла service.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 service.xml для сервиса терминалов определяется **authType="arcusid"**, а для ККМ **authType="ip"**. Пример такой конфигурации ниже:

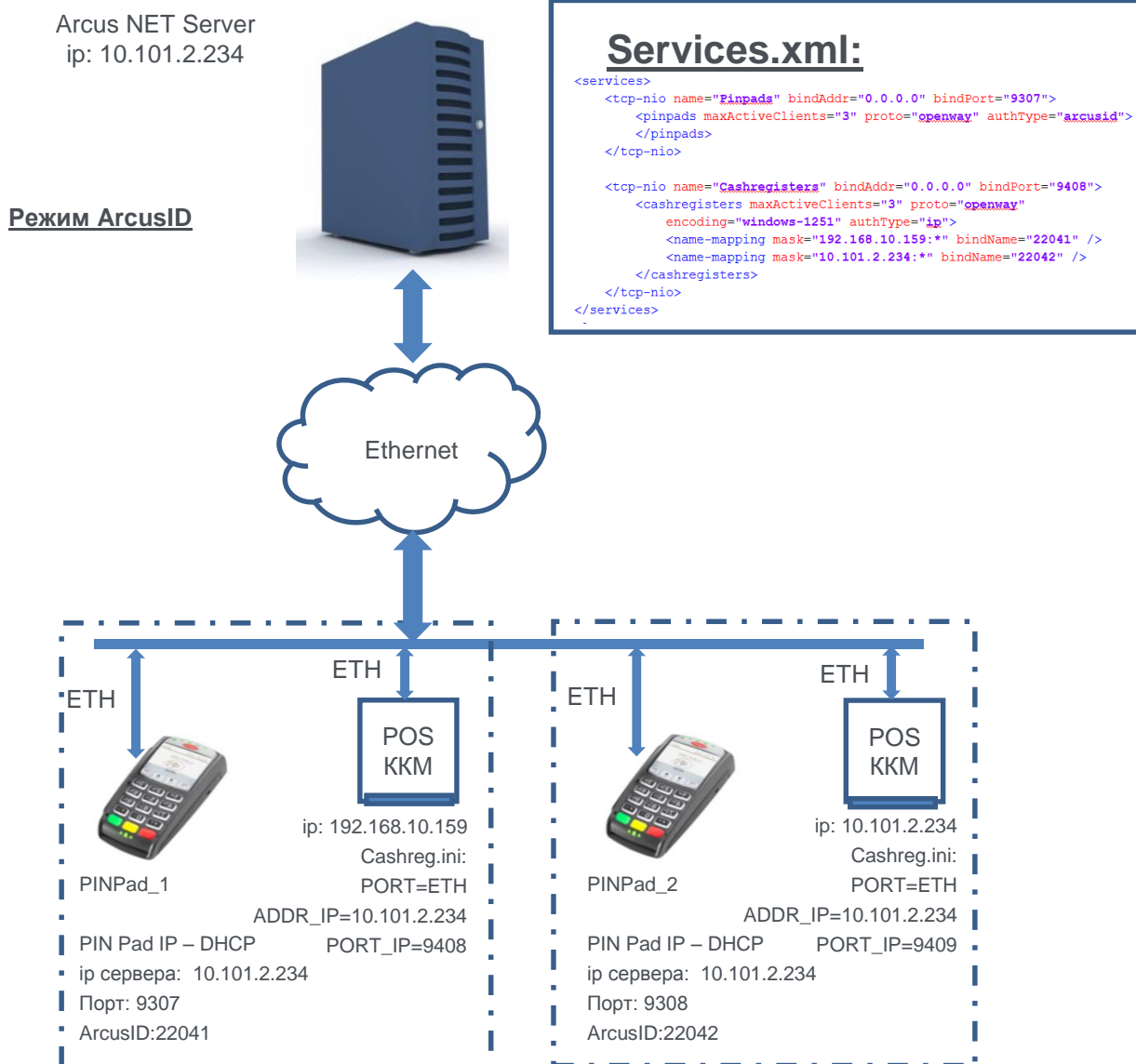


Рис. 14. Пример настройки Arcus NET Server v.2 – Режим Arcus ID

<?xml version="1.0" encoding="UTF-8"?>

```

<configuration>
  <services>
    <tcp-nio name="Pinpads" bindAddr="0.0.0.0" bindPort="9307">
      <pinpads maxActiveClients="3" proto="tieto" 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

Arcus NET Server
ip: 10.101.2.234

Режим ip

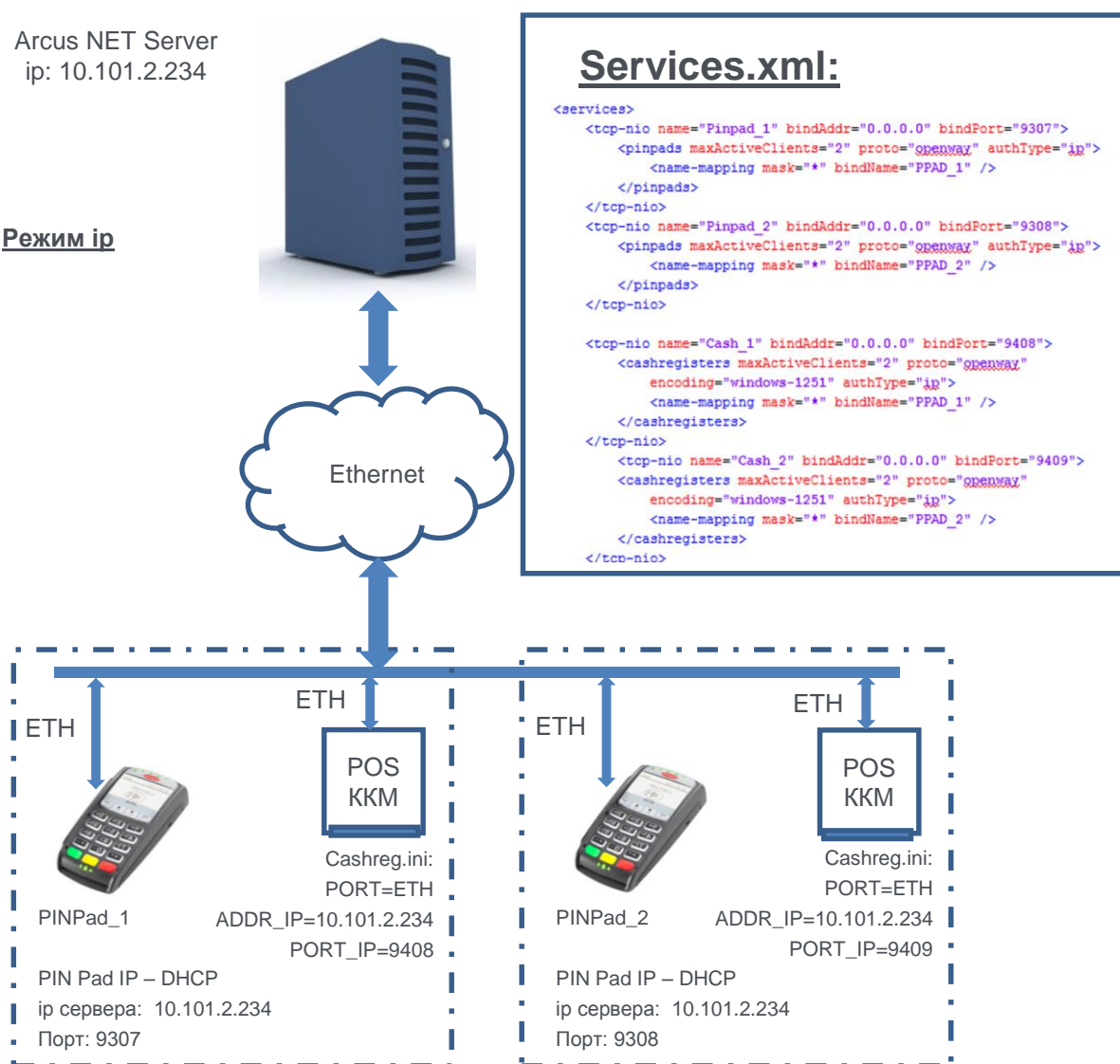


Рис. 15. Пример настройки Arcus NET Server v.2 – Режим IP

```
<?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>
    <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>

```

6_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


6_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>
</sslProfiles>
```

6_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:

	Это касается чека на возврат в 1С: Управлении небольшой фирмой ред.1.3
---	---

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

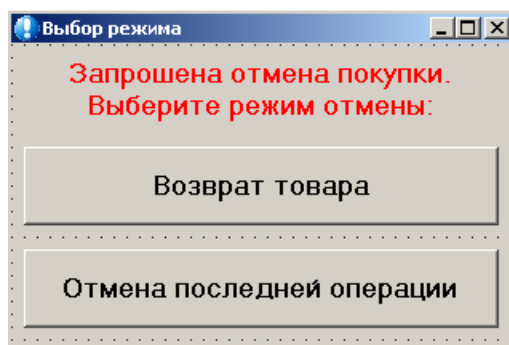


Рис. 16. Выбор режима

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

7_Основные этапы установки и настройки ARCUS2 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 ARCUS2.
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».

7_1 Тестовая транзакция для ОС Windows

Проведите тестовую транзакцию, запустив **CommandLineTool.exe**:

Формат запуска **CommandLineTool.exe** следующий:

CommandLineTool.exe /o[Код_операции_ККМ] /с[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.exe /o1 /с643 /a12345
```

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

7_2 Тестовая транзакция для ОС DOS

Проведите тестовую транзакцию, запустив CASHREG.EXE:

Формат запуска **cashreg.exe** следующий:

cashreg.exe Код_операции_ККМ *Сумма, где

Код операции ККМ – выбранная кассиром операция; (символ перед знаком '=' в строке файла **ops.ini**).

Сумма – сумма операции в минимальных единицах валюты, без разделителя.

Например:

```
C:\> cashreg.exe 1 12345
```

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

7_3 Тестовая транзакция для ОС Linux

Проведите тестовую транзакцию, запустив **Cashreg** с атрибутами в следующем примере:

Формат запуска **Cashreg** следующий:

Cashreg /o[Код_операции_ККМ] /с[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:

```
$ ./Cashreg /o1 /с643 /a12345
```

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

Приложение 1. Интеграция с использованием COM Object

Для работы с Connector рекомендуется следующая последовательность действий:


1. Создать Request экземпляр объекта *ISAPacketObj* для пересылки данных Server для проведения транзакции.
2. Создать Response экземпляр объекта *ISAPacketObj* для получения результатов транзакции от Server.
3. Создать экземпляр объекта *ICconnector Obj* для организации каналов обмена данными с Server.
4. Подготовить данные для проведения транзакции, заполнив соответствующие поля в *Request*.
5. Послать запрос на проведение транзакции вызовом метода *Exchange*.
6. Повторить предыдущий пункт столько раз, сколько транзакций требуется провести.

Описание PCPOSTConnectorObj

Метод Exchange

1. Обменяться информацией с Server, т.е. отправить серверу запрос на проведение транзакции и получить от него ответ.
2. Вызвать Connector. Обменяться данными (*Request*, *Response*, *Timeout*, *Result*)
 - *Request* – Данные для проведения транзакции;
 - *Response* – Результат проведения транзакции;
 - *Timeout* – Таймаут на получение результатов проведения транзакции; (не используется)
 - *Result* – значение, возвращаемое методом *Exchange*.

Свойство	№ поля	Описание поля
Amount	0	Сумма операции, выраженная в минимальных единицах валюты
CurrencyCode	4	Код валюты операции
DateTimeHost	6	Оригинальная дата и время совершения операции YYYYMMDDHHMMSS на хосте
CardEntryMode	8	Способ ввода карты (возможные варианты: 1 – ручной, 2 – считыватель магнитной полосы, 3 – микропроцессорная карта вводится через микропроцессорный считыватель ПИН-пад)
PAN	10	Номер карты (заполняется совместно с полем 11, В этом случае
CardExpiryDate	11	Срок действия карты YYMM (заполняется совместно с полем 10, в этом случае поле 12 не заполняется)
TRACK2	12	Данные дорожки 2 (в этом случае поле 10 и 11 не заполняются)
AuthorizationCode	13	Код авторизации (может возвращаться хостом)
ReferenceNumber	14	Номер ссылки
ResponseCodeHost	15	Код ответа
TextResponse	19	Дополнительные данные ответа
DateTimeCRM	21	Дата и время проведения операции на терминале GGGMMDDЧчммсс
TrxID	23	Идентификатор транзакции в коммуникационном сервере
OperationCode	25	Код операции
TrxIDCRM	26	Уникальный номер транзакции
CRMID	27	Идентификатор терминала на входе

Свойство	№ поля	Описание поля
SLIP		<p>Образ чека передается в объект response метода Exchange класса PCPOSTConnectorObj. Образ передается в кодировке указанной в параметре PRINTERCHARSET в cashreg.ini.</p> <p>Образ чека также сохраняется в файл. В случае, если с пин-пада не приходили команды на печать, свойство имеет значение <i>null</i>.</p> <div>  <p>Файл из СОМ-объекта формируется ТОЛЬКО при печати на кассе! Доступно начиная с версии от 2014\12\25</p> </div>
CommodityCode		Код товара (тэг "\x1F\x1F")
PaymentDetails		Детали транзакции (req тэг "\x1F\x4E", rsp тэг "\x1F\x26")
ProviderCode		Код провайдера (тэг "\x1F\x4F")
ProcessingFlag		PIN флаг (тэг "\x1F\x61")
CommandResult		Код ответа RESPONSE_CODE
CVV2		CVV2 (тэг "\x1F\x5D")
TerminalOutID		Идентификатор терминала (тэг "\x9F\x1C")
CashbackAmount		cashback - сумма операции, в минимальных денежных единицах (тэг "\x9F\x03")
RFU		Тоже самое поле, что и TRACK2. Сделано для поддержки не PA_DSS интерфейсов. (тэг "\x57")
BIN		Первые 6 цифр PAN
Last4Digits		Последние 4 цифры PAN
Hash		Хэш (SHA1) от PAN
EncData		Шифрованные данные карты (тэг "\xA1\xA1")
Signature		Запуск механизма захвата подписи на терминале
SignaturePath		Установить путь для сохранения файлов подписей
TerminalIDX		Индекс для выбора терминала (USEJL)

Для использования данных объектов необходимо создать в проекте ссылку на СОМ объект.

Приложение 2. Настройка ПИН-пад

Перед подключением к ККМ необходимо проверить настройки ПИН-пад:

1. Перезагрузить пин-пад одновременным нажатием клавиш UP+DOWN+STOP (если используется модель 3070, то отключить и включить питание).
2. Во время загрузки нажать и удерживать клавишу MENU (на некоторых моделях данная клавиша обозначена пиктограммой «квадрат»).
3. На вопрос «Работа с кассой?» нажать ОК (выбрать ДА в меню выбора режима).
4. Выбрать настройки СОМ-порта ПИН-пад (номер, скорость, бит данных, стоп бит).
5. Выбрать настройки IP соединения с сервером. На вопрос «IP через кассу?» нажать ОК=ДА или STOP=НЕТ
6. На вопрос «Диалог на кассе?» нажать ОК, если предполагается вывод диалогов на экран кассы или STOP, если диалоги предполагаются на экране ПИН-пад.
7. Выбрать настройки принтера. На вопрос «Принтер на кассе?» нажать ОК=ДА или STOP=НЕТ
8. Ввести ширину чека (зависит от типа ПО ПИН-пад и используемого ФРК).
9. Выбрать настройку для операции РАСЧЕТ. На вопрос «Код авторизации с кассы?» нажать ОК=ДА или STOP=НЕТ

Приложение 3. Описание структуры

UserAuthInt

```
typedef struct
{
    int handle;
    int abg_id;
    int operType;
    char track2[InTrack2]/rfu[InRFU];
    char pan[InPan];
    char expiry[ InExpiry ];
    char pay_acc[InPan];
    char additional_payment_data[80];
    char amount[ InAmount ];
    char original_amount[ InAmount ];
    char currency[ InCurrency ];
    char terminalID[InTerminalID ];
    char rrn[InRetrievalReference ];

    char authCode[ InAuthIdentResponse ];
    char responseCode[InResponseCode ];
    char cardType[InCardType];
    char date[InDate];
    char time[InTime];
    char payment_data[InPayData];
    char data_to_print[InPayData];
    char home_operator[InPayData];
    char received_text_message[InReceivedTextMsg];
    char text_message[InReceivedTextMsg];
    char AID[InAID];
    char ApplicationLabel[InApplicationLabel];
    char TVR[InTVR];
    int system_res;
    char enc_data[64];

    char cardholder_name[InCardholderName];
    char Balance[InAmount+1];
}UserAuthInt;
Enum {
    InProcCode = 7,
    InTrack2 = 60,
    InPan = 20,
    InExpiry = 5,
    InAmount = 13,
    InCurrency = 4,
    InTerminalID = 9,
    InSpdhTerminalID=17,
    InRetrievalReference = 13,
    InAuthIdentResponse = 9,
    InResponseCode = 4,
    InCardType = 80,
    InDate = 7,
    InTime = 7,
    InBatchNum = 8,
    InTrack2Credit = 50,
    InPinblock = 17,
    InPayData = 50,
    InPayId = 3,
    InMtid = 5,
    InReceivedTextMsg=80,
    InAID=80,
    InApplicationLabel=80,
    InTVR=80,
    InRFU=60
};

//[in] Код операции (кассовый)
//[in] Трек2
//[out] PAN
//[out] Expiry Date ГГММ
//не используется
//не используется
//[in] Сумма в копейках
//[in] Оригинальная сумма в копейках
//[in] Код валюты
//[out][in] ID терминала
//[in][out] Ссылка (только для операций для которых нужно, в остальных случаях - пуста)
//[out][in] Код авторизации
//[out] Код ответа
//[out] Название типа карты
//[out] Дата транзакции
//[out] Время транзакции
//не используется
//не используется
//не используется
//[out] Расшифровка
//[out] EMV AID
//[out] EMV ApplicationLabel
//[out] EMV TVR
//не используется
// [in][out]шифрованные данные карты(PAN) (для ПО с функцией поддержки HRS)
```

UserAuthintFull



Структура доступна с декабря 2015 года.

```
typedef struct
{
    int handle;
    int abg_id;
    int operType;
    char rfu[InRFU];
    char pan[InPan];
    char expiry[ InExpiry ];
    char pay_acc[InPan];
    char additional_payment_data[80];
    char amount[ InAmount ];
    char original_amount[ InAmount ];
    char currency[ InCurrency ];
    char terminalID[InTerminalID ];
    char rrn[InRetrievalReference ];

    char authCode[ InAuthIdentResponse ];
    char responseCode[InResponseCode ];
    char cardType[InCardType];
    char date[InDate];
    char time[InTime];
    char payment_data[InPayData];
    char data_to_print[InPayData];
    char home_operator[InPayData];
    char received_text_message[InReceivedTextMsg];
    char text_message[InReceivedTextMsg];
    char AID[InAID];
    char ApplicationLabel[InApplicationLabel];
    char TVR[InTVR];
    int system_res;
    unsigned short receiptNumber;
    unsigned short Invoice;
    int operatopn_type;
    char card_number[20];
    char merchant[80];
    char date_buffer[100];
    unsigned short debit_number;
    char debit_total[7];
    char debit_clear_total[7];
    unsigned short return_number;
    char return_total[7];
    char return_clear_total[7];
    unsigned long butch_num;
    char TrAmount[13];
    char enc_data[64];

    char cardholder_name[InCardholderName];
    char max_discount[InAmount];
    char min_discount[InAmount];
    char commission_amount[ InAmount+1 ];
    char PaymentsData[102];

    char BIN[7];
    char Hash[41];
    char Last4Digits[7];
    char isDiscountedGoods;
    char Balance[InAmount+1];

    // [in] Код операции (кассовый)
    // [out] PAN
    // [out] Expiry Date ГГММ
    // не используется
    // не используется
    // [in] Сумма в копейках
    // [in] Оригинальная сумма в копейках
    // [in] Код валюты
    // [out][in] ID терминала
    // [in][out] Ссылка (только для операций для которых нужно, в остальных случаях - пуста)
    // [out][in] Код авторизации
    // [out] Код ответа
    // [out] Название типа карты
    // [out] Дата транзакции
    // [out] Время транзакции
    // не используется
    // не используется
    // не используется
    // [out] Расшифровка
    // [out] EMV AID
    // [out] EMV ApplicationLabel
    // [out] EMV TVR
    // не используется
    // номер чека
    // номер платежа
    // тип операции
    // номер карты
    // Идентификатор организации
    //
    // количество дебетовых операций
    // итог дебет
    // итог дебет без учета комиссий и скидок
    // Количество возвратов
    //
    //
    // номер смены / пакета
    //
    // [in][out] зашифрованные данные карты (PAN) (для ПО с функцией поддержки HRS)
    // имя владельца карты
    // максимальная сумма скидки (в минимальных единицах валюты)
    // минимальная сумма скидки (в минимальных единицах валюты)
    // сумма комиссии (в минимальных единицах валюты)
    // для unipay (каждое поле должно заканчиваться «NULL»)
    PaymentsData [0:2] – Идентификатор платежа;
    PaymentsData [3:35] – Дополнительные данные 1;
    PaymentsData [36:68] - Дополнительные данные 2;
    PaymentsData [69:101] - Дополнительные данные 3;
    // BIN карты, поле должно заканчиваться «NULL»
    // Хэш номера карты (SHA1)
    // Последние 4 цифры номера карты
    // Признак дисконтных товаров в чеке
    // Баланс карты
```


char TraceID[InTraceID];	//идентификатор предавторизации
char OriginalDateTime[InDateTime];	//Дата и время оригинальной операции
unsigned short CashReceiptNumber;	//Номер кассового чека
char OutPutTransactionData[256];	//Дополнительные данные транзакции (сообщение от хоста)
char TransactionID[19];	//Идентификатр транзакции
char ComissionOffline[InAmount];	//Офлайн комиссия
char ComissionOfflineCurrency[InCurrency];	//Валюта офлайн комиссии
char ComissionAcquirer[InAmount];	//Комиссия эквайера
char ComissionAcquirerCurrency[InCurrency];	//Валюта комиссии эквайера
char ComissionIssuer[InAmount];	//Комиссия эмитента
char ComissionIssuerCurrency[InCurrency];	//Валюта комиссии эмитента
char AuthorizedAmount[InAmount];	//Авторизованная сумма
char AuthorizedAmountCurrency[InCurrency];	//Валюта авторизованной суммы

```

}UserAuthIntFull;
Enum {
InProcCode = 7,
InRFU = 60,
InPan = 20,
InExpiry = 5,
InAmount = 13,
InCurrency = 4,
InTerminalID = 9,
InSpdhTerminalID=17,
InRetrievalReference = 13,
InAuthIdentResponse = 9,
InResponseCode = 4,
InCardType = 80,
InDate = 7,
InTime = 7,
InBatchNum = 8,
InRFUCredit = 50,
InPinblock = 17,
InPayData = 50,
InPayId = 3,
InMtid = 5,
InReceivedTextMsg=80,
InAID=80,
InApplicationLabel=80,
InTVR=80,
InCardholderName = 27,
InTraceID = 61,
InDateTime = 13
};

```

Приложение 4. Пример использования ArcCom.dll

Вызов UserAuthint

```
// CASHREG_DLL_TEST.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <windows.h>
// #include <iostream.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <windows.h> #include <winbase.h>
enum {
InProcCode = 7,
InTrack2 = 60,
InPan = 20,
InExpiry = 5,
InAmount = 13,
InCurrency = 4,
InTerminalID = 9,
InSpdhTerminalID=17,
InRetrievalReference = 13,
InAuthIdentResponse = 9,
InResponseCode = 4,
InCardType = 80,
InDate = 7,
InTime = 7,
InBatchNum = 8,
InTrack2Credit = 50,
InPinblock = 17,
InPayData = 50,
InPayId = 3,
InMtid = 5,
InReceivedTextMsg=80,
InAID=80,
InApplicationLabel=80,
InTVR=80
};

typedef struct
{
int handle;
int abg_id;
int operType;
char track2[InTrack2];
char pan[InPan];
char expiry[ InExpiry ];
char pay_acc[InPan];
char additional_payment_data[80];
char amount[ InAmount ];
char original_amount[ InAmount ];
char currency[ InCurrency ];
char terminalID[ InTerminalID ];
char rrn[ InRetrievalReference ];

char authCode[ InAuthIdentResponse ];
char responseCode[ InResponseCode ];
char cardType[InCardType];
char date[InDate];
char time[InTime];
char payment_data[InPayData];
char data_to_print[InPayData];

//[in] Код операции (кассовый)
//[in] Трек2
//[out] PAN
//[out] Expiry Date ГГММ
//не используется
//не используется
//[in] Сумма в копейках
//[in]Оригинальная сумма в копейках
//[in] Код валюты
//[out] ID терминала
//[in][out] Ссылка (заполнять для операций, где
        нужно, в остальных случаях - пусто)
//[in][out] Код авторизации
//[out] Код ответа
//[out] Название типа карты
//[out] Дата транзакции
//[out] Время транзакции
//не используется
//не используется
}
```

```

char home_operator[InPayData]; //не используется
char received_text_message[InReceivedTextMsg]; //не используется
char text_message[InReceivedTextMsg]; //out] Расшифровка
char AID[InAID]; //out] EMV AID
char ApplicationLabel[InApplicationLabel]; //out] EMV ApplicationLabel
char TVR[InTVR]; //out] EMV TVR
int system_res; //не используется
// ТОЛЬКО ДЛЯ МОДУЛЯ АРКУС2 С ФУНКЦИОНАЛОМ HRS
char enc_data[64]; //in][out] шифрованные данные карты (PAN)
}UserAuthInt;
#define GET_DLL_FUNC(x,y) x=(y)GetProcAddress((HMODULE)hLib,#x);if(x==NULL)
{printf("Unable to load %s function! %x (%lu)",#x,GetLastError(),
GetLastError());getch();return -1;}
typedef int (__cdecl *_PROCFUNC) (UserAuthInt *);
_PROCFUNC ProcessOw;
HMODULE hLib;
int LoadOWDll(void){
hLib=LoadLibrary("c:\\Arcus2\\dll\\arccom.dll");
if(hLib==NULL)
{
printf("Unable to load library!");
getch();
return -1;
}
GET_DLL_FUNC(ProcessOw,_PROCFUNC);
return 0;
}
#define PRINTF_VAR(v,t) printf("#v" = "t\\n",v)
int _tmain(int argc, _TCHAR* argv[])
{
int res;
if(LoadOWDll()>=0){
UserAuthInt TestTrans;
int result=0;
memset(&TestTrans,0,sizeof(UserAuthInt));
TestTrans.operType=1; // ОПЛАТА
sprintf(TestTrans.currency,"643"); //Рубли
sprintf(TestTrans.amount,"123"); // 1 руб 23 коп
result=ProcessOw(&TestTrans);
PRINTF_VAR(result,"%i");
PRINTF_VAR(TestTrans.responseCode,"%s");
PRINTF_VAR(TestTrans.text_message,"%s");
PRINTF_VAR(TestTrans.currency,"%s");
PRINTF_VAR(TestTrans.amount,"%s");
PRINTF_VAR(TestTrans.rn,"%s");
PRINTF_VAR(TestTrans.authCode,"%s");
getchar();
libend:
FreeLibrary(hLib);
}
return 0;
}

```

Вызов UserAuthIntFull

Перед использованием структуры ее необходимо обнулить функцией memset().

```

#include <windows.h>
#include <iostream>
#include <string>
enum
{
InProcCode = 7,

```

```

InRFU = 60,
InPan = 20,
InExpiry = 5,
InAmount = 13,
InCurrency = 4,
InTerminalID = 9,
InSpdhTerminalID = 17,
InRetrievalReference = 13,
InAuthIdentResponse = 9,
InResponseCode = 4,
InCardType = 80,
InDate = 7,
InTime = 7,
InBatchNum = 8,
InRFUCredit = 50,
InPinblock = 17,
InPayData = 50,
InPayId = 3,
InMtid = 5,
InReceivedTextMsg = 80,
InAID = 80,
InApplicationLabel = 80,
InTVR = 80,
InCardholderName = 27,
InTraceID = 61,
InDateTime = 13
};

```

```

#ifdef _MSC_VER
#pragma pack(push, 1)
#endif

```

```

typedef struct
{
int handle;
int abg_id;
int operType;
char rfu[InRFU];
char pan[InPan];
char expiry[InExpiry];
char pay_acc[InPan];
char additional_payment_data[80];
char amount[InAmount];
char original_amount[InAmount];
char currency[InCurrency];
char terminalID[InTerminalID];
char rrn[InRetrievalReference];
char authCode[InAuthIdentResponse];
char responseCode[InResponseCode];
char cardType[InCardType];
char date[InDate];
char time[InTime];
char payment_data[InPayData]; // data to send back to host
char data_to_print[InPayData]; // data to print on receipt
char home_operator[InPayData]; // data to print on receipt
char received_text_message[InReceivedTextMsg];
char text_message[InReceivedTextMsg];

```

```

char AID[InAID];
char ApplicationLabel[InApplicationLabel];
char TVR[InTVR];
int system_res;
unsigned short receiptNumber;
unsigned short Invoice;
int operatopn_type;
char card_number[20];
char merchant[80];
char date_buffer[100];
unsigned short debit_number;
char debit_total[7];
char debit_clear_total[7];
unsigned short return_number;
char return_total[7];
char return_clear_total[7];
unsigned long butch_num;
char TrAmount[13];
char enc_data[64];
char cardholder_name[InCardholderName];
char max_discount[InAmount];
char min_discount[InAmount];
char commission_amount[InAmount + 1];
char PaymentsData[102]; //in unipay (each field must be NULL terminated) PaymentsData [0:2] - PaymentID;
PaymentsData [3:35] - PayFieldValue1; PaymentsData [36:68] - PayFieldValue2; PaymentsData [69:101] -
PayFieldValue3;
char BIN[7];
char Hash[41];
char Last4Digits[7];
char isDisoutedGoods;
char Balance[InAmount + 1];
char TraceID[InTraceID];
char OriginalDateTime[InDateTime];
unsigned short CashRecieptNumber;
char OutPutTransactionData[256];
char TransactionID[19];
char ComissionOffline[InAmount];
char ComissionOfflineCurrency[InCurrency];
char ComissionAcquirer[InAmount];
char ComissionAcquirerCurrency[InCurrency];
char ComissionIssuer[InAmount];
char ComissionIssuerCurrency[InCurrency];
char AuthorizedAmount[InAmount];
char AuthorizedAmountCurrency[InCurrency];
}UserAuthIntFull;

#ifdef _MSC_VER
#pragma pack(pop)
#endif

```

```

typedef int(__cdecl *_PCPOSFUNCFULL)(UserAuthIntFull *auth_st_full, int size);

```

```

int main(void) {
HMODULE hLib =
LoadLibrary("D:\\work\\workspaces\\arcus_workspace\\arccom\\ArcCom___Win32_Debug_MinDependency\\
ArcCom.dll");

```

```
_PCPOSFUNCFULL ProcessOwFull = (_PCPOSFUNCFULL)GetProcAddress(hLib, "ProcessOwFull");
UserAuthIntFull auth;
memset(&auth, 0, sizeof(UserAuthIntFull));
auth.operType = 1;
strncpy_s(auth.amount, sizeof(auth.amount), "1234", 4);
int res = ProcessOwFull(&auth, sizeof(UserAuthIntFull));
FreeLibrary(hLib);
return 0;
}
```

Приложение 5. 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 из структуры UserAuthInt можно получить RRN, желательно его и использовать для операций «Универсальная отмена» и «Возврат».



Использование каждой из них необходимо обговаривать непосредственно с банком.

Приложение 6. Функционал DLS (опционально)

Данный функционал доступен для процессинга OpenWay.

Функционал реализует применение системы лояльности и фискализации DLS – поддержка протокола АПДК версии 8.1.2.

Для подключения функционала необходимо обновить ArcCom.dll и CommandLineTool.exe.

Поддерживается 2 режима интеграции с кассой: полная и легкая интеграции.

Режим легкой интеграции с кассой

Обеспечивает работу системы лояльности без модернизации ПО кассы.

Основан на использовании специализированных кодов валют (для операций Оплата / Возврат).

- 999 - будет выполнена операция Оплата за бонусы / Возврат за бонусы
- 998 - будет выполнена операция Оплата за наличные с начислением бонусов / Возврат за наличные со списанием бонусов.

Режим полной интеграции с кассой

Оплата

Стандартный запуск операции.

В режиме работы с кассой опционально поддерживается передача на хост DLS списка товаров.

Для этого необходимо заполнить структуру UserAuthInt и передать ее функции:

```
DisProcessOw(UserAuthInt *auth_st, char * data, unsigned int data_size);
```

где data - указатель на список товаров.

Оплата наличными с начислением бонусов и без

Аналогично операции оплата +

В режиме работы с кассой поддерживает режим Silent, когда не запрашивается на предъявление карта и не печатается чек.

Этот режим включается записью '1' в поле PaymentsData[102] структуры UserAuthInt.

Возврат со списанием бонусов и без

Операция возврат подразумевает передачу на терминал:

- список товаров (аналогично операции оплата)
- rpn (поле rpn структуры UserAuthInt)

При отсутствии rpn будет выведен диалог для ввода rpn. Если терминал не смог найти по rpn дату проведения операции, будет выведен диалог для ввода даты.

Открытие смены

Подразумевает передачу на терминал:

- код кассира (поле rpn структуры UserAuthInt)
- (опционально) фио кассира (поле PaymentsData[102] структуры UserAuthInt)

При отсутствии кода кассира будет выведен диалог для ввода кода.

Инкассация

Подразумевает передачу на терминал:

- внесенная сумма (поле amount структуры UserAuthInt)
- изъятая сумма (поле original amount структуры UserAuthInt)

При отсутствии данных будет выведен диалог для ввода.

Приложение 7. Информация о документе

Список таблиц

/	Таблица 1. Хронология изменений	5
/	Таблица 2. Условные обозначения	6
/	Таблица 3. Глоссарий.....	6
/	Таблица 4. Аббревиатуры	8
/	Таблица 5. Атрибуты tcp-nio	48
/	Таблица 6. Атрибуты тэгов pinpads и cashregisters	48
/	Таблица 7. Атрибуты FileService	49
/	Таблица 8. Атрибуты тэгов fileTransactions	49
/	Таблица 9. Атрибуты тэгов FileLogger и asyncFileLogger	50

Список рисунков

Рис. 1. Схема работы ARCUS 2 CAP	10
Рис. 2. SSL шифрование в режиме IP через кассу.....	11
Рис. 3. Схема взаимодействия кассового ПО с модулем Arcus2.....	18
Рис. 4. Банковский чек.....	19
Рис. 5. Чек с ошибкой	19
Рис. 6. Виртуальная клавиатура	28
Рис. 7. Схема работы ARCUS 2 HRS CAP	34
Рис. 8. Схема работы Модуля AC1	35
Рис. 9. Выбор режима	57
Рис.10. Общая схема взаимодействия.....	38
Рис. 11. Схема работы ARCUS NET Server v.1.3.....	39
Рис. 12. Пример настройки сервера 1 ККМ-1 ПИН-пад.....	42
Рис. 13. Пример настройки сервера (более одной ККМ (ARCUS2) и более одного ПИН-пад)	43
Рис. 14. Схема работы ARCUS NET Server v.2.0.....	44
Рис. 15. Пример настройки Arcus NET Server v.2 – Режим Arcus ID	52
Рис. 16. Пример настройки Arcus NET Server v.2 – Режим IP	53