

Программное Обеспечение “Тагра- Управления паркоматом”

(ParkCashDesk2)

**Функциональные
характеристики**



Содержание

1. Общие сведения.....	4
1.1. Основные функциональные характеристики.....	4
2. Файлы конфигурации.....	4
2.1. Описание конфигурационного файла DevicesSet.xml	4
2.2. Описание клавиш клавиатуры.....	9

1. Общие сведения

1.1. Основные функциональные характеристики

ПО обеспечивает следующий функционал в составе Паркомата:

- 1) Прием оплаты безналичным способом с выбором пользователем зоны парковки, государственного номера транспортного средства (ТС), времени парковки, опционально тип ТС.
- 2) Расчет стоимости парковки в зависимости от выбранных пользователем параметров парковки. Стоимость может рассчитываться локально при помощи гибко-настраиваемого калькулятора тарифов, так и получать online при помощи стороннего калькулятора.
- 3) Прием оплаты банковскими картами на терминалах Yarus K2100 и других работающих по протоколу ТТК2 (Ingenico iSelf с ПО Сбербанка)
- 4) Прием оплаты транспортными картами казани
- 5) Прием оплаты prepaid картами парковочного пространства СПб
- 6) Прием оплаты prepaid картами парковочного пространства г.Москвы
- 7) Оплата наличными денежными средствами совместимость по устройствам по протоколу CCNET для купюроприемников и по протоколу CCTalk для монетопримников.
- 8) Поддержка спящего режима.
- 9) Поддержка сенсорного экрана и емкостной клавиатуры.

2. Файлы конфигурации

В данном разделе описываются структура конфигурационных файлов ПО и некоторые важные параметры. Как ранее было отмечено вся конфигурация осуществляется при обновлении. Этот раздел помогает лучше ориентироваться в ПО при обслуживании и делать какие то ручные правки для проверки поведения и дальнейшего внесения правок в обновление.

2.1. Описание конфигурационного файла DevicesSet.xml

Это основной файл конфигурации приложения. Он находится в его рабочем каталоге /data/terminalapp и имеет формат xml файла. Как ранее было указано

приложение строится по принципу устройство-ориентированного, исходя из этого конфигурация состоит из следующих основных разделов:

/DevicesSet/Devices – описание и конфигурация всех устройств которые будут проинициализированы в приложении.

/DevicesSet/DevicesTree – Описание зависимостей устройств между собой. Используется при проверке устройств. Все устройства, которые указаны в первом разделе должны быть упомянуты в данном разделе, иначе устройства не будут проинициализированы.

/DevicesSet/DevicesManager – содержит настройки менеджера устройств в настоящий момент содержит единственный параметр CheckingThreadPoolSize определяющий количество потоков для выполнения проверки устройств.

Первый раздел состоит из набора тегов Device представляющих конфигурацию каждого из устройств. В общем виде теги имеют вид:

```
<Device Id="1048576" Type="Monitoring" Name="Monitoring">
```

@Id – идентификатор устройства. Для корректной работы должен совпадать с идентификаторами устройств в системе мониторинга.

@Name – Видимое имя которое будет отображаться в сервисном меню. Возможные имена, которые настраиваются в обновлении были указаны ранее.

@Type – функциональный тип устройства в приложении. Возможные типы следующие:

AppModule – определяет логику приложения используется как устройство олицетворяющее все устройство

Monitoring – устройство олицетворяет логику взаимодействия с сервером обновления.

FiscalRegistrar – фискальный регистратор.

Printer – печатающее устройство. Если устройство совмещено с фискальным регистратором то указывается два типа через запятую.

PowerInterrupter – устройство для заряда батарей

PowerMeter – счетчик электроэнергии

Sensor – вход (сухой контакт)

BankCardTerminal – банковский терминал

Keyboard – устройство читающее статус клавиатуры

ScriptDevice – устройство получающие данные из скрипта

TariffDevice – сервис расчета тарифа. Позволяет отражать состояние для некоторых внешних калькуляторов тарифов.

PaymentSystem – виртуальное устройство символизирующее платежную систему включающую работу со всеми платежными и фискальными устройствами

PaymentDevice – устройство принимающее оплату

NfcReader – считыватель бесконтактных карт

CardProcessing – виртуальное устройство определяющее логику обслуживания карт определенного процессинга предоплаченных карт

Каждое из устройств также содержат следующие теги в конфигурации:

ServiceMenuSet/LockKeyId – код клавиши назначенной для блокировки\разблокировки устройства в сервисном меню.

CheckInTechMode (boolean) – включает или отключает проверку устройства в техническом режиме. Техническим режимом называется режим когда устройство работает в энергосберегающем режиме и выходит из этого режима не для обслуживания пользователя, а для отправки на сервер мониторинга информации о своем состоянии.

SelfcheckDelay – период в мс определяющий задержку в течении которой устройство будет не будет полагаться непригодным для использования при не прохождении проверки. Период отсчитывается от момента запуска приложения или выхода из энергосберегающего режима. Этот параметр позволяет устройствам загрузиться, не показывая ошибки пользователю и давая ему работать в интерфейсе вводя основные параметры своей парковки.

SelfcheckPeriod – период в мс через который устройству будет назначен очередной цикл проверки.

NetSettings/Address – для сетевых устройств содержит ip адрес устройства

NetSettings/Port – для сетевых устройств содержит ip порт устройства

ComSettings/PortName – для устройств подключенных через com-порт содержит имя порта

ComSettings/Baudrate – для устройств подключенных через com-порт содержит настройку скорости подключения

AccessSem/@Name – имя семафора блокировки. Используется для разграничения доступа к устройствам использующих общий ресурс доступа. Например если два устройства подключены на один порт по шине RS485, то при указании семафора с именем RS485 у этих двух устройств они не смогут быть проверены или захвачены для работы одновременно из разных потоков проверки или основного рабочего потока.

Driver – содержит имя класса драйвера который будет загружен для данного устройства.

Специализированные конфигурационные параметры устройства TheDevice (Тип AppModule).

Для данного приложения тег Driver в секции AppModule должен быть установлен в значение ParkomatTerminal.

Basedir – базавый каталог для относительных путей

TheDeviceNumber – номер устройства.

DeviceAddress - Адрес нахождения паркомата. Используется для экрана информации о парковке и для использования в чеке.

SoftUpdaterScript – команда которая будет выполняться при получении команды обновиться с сервера. Доступно указание переменных, которые могут быть заполнены данными от сервера: {\$parkomatNum} – номер паркомата,

{\$UpdateSoftPath} – требуемая версия, {\$SetupParams} – дополнительные параметры для установки.

PackLogsScript – команда которая будет выполняться при получении команды с сервера упаковать и отправить логи. Доступно указание переменной {\$UploadLogsPath}. В контексте данная переменная означала http путь для отправки логов, но в последствии доработок сервера направляется дата за которую необходимо выгрузить логи в формате YYYYMMDD.

SessionsDir – каталог временного хранения данных сессии пользователя.

PowerSaveMode – режим экономии энергии. Принимает значения suspend или none.

NoActionTimeout_msec – Время в мс, после которого пользовательская сессия без подтверждения активности пользователя перезапускается и ПО паркомата переходит в режим ожидания пользователя.

FinishingActivityTimeout_msec - Время в мс ожидания подтверждения активности на экране подтверждения активности до ухода в сон.

FinishingTransactionTimeout_msec – время в мс задержки вывода экрана результата операции

SrvMenuAttemptsToCancelFailedTransaction, SrvMenuShowHintKeyId,
SrvMenuScrollDownKeyId, SrvMenuScrollUpKeyId, SrvMenuCloseMenuKeyId,
SrvMenuCloseAppKeyId, SrvMenuShutdownKeyId, SrvMenuRestartAppKeyId,
SrvMenuPrintFailedTransactionKeyId, SrvMenuCancelFailedTransactionKeyId,
SrvMenuSummaryReportKeyId, SrvMenuOpenBankTerminalMenuKeyId,
SrvMenuDeleteFailedTransactionKeyId – настройки кодов клавиш в сервисном меню

SrvMenuStopAppCmd – Команда для останова приложения. Вызывается по нажатию кнопки останова приложения из сервисного меню

```
<SrvMenuSwitchTabKeyId>16777219</SrvMenuSwitchTabKeyId>
```

```
<SrvMenuStopAppCmd>/etc/init.d/terminalapp stop 2>/dev/null 1>/dev/null  
0</dev/null &</SrvMenuStopAppCmd>
```

WaitForParkingCardTimeout_msec - Время ожидания транспортной карты на экране проверки баланса карты

PublishZeroPaymentsInParknow - Делать попытку отправки транзакции старта парковки в процессинг ParkNow при нулевой сумме

ServiceMenuTranslations – путь к файлу переводов сервисного меню

AppThreadInfoFile – файл активности основного потока ПО. Позволяет настроить WatchDog на зависание ПО.

TaxGroup=0, DepartmentNum=0, PaymentTypeSign=4, PaymentItemSign=4 – фискальные параметры

Специализированные конфигурационные параметры устройства Monitoring (тип Monitoring).

Для данного приложения тег `Driver` в секции `AppModule` должен быть установлен в значение `ParkomatMonitoring`.

`RC2Key` - Ключ шифрования при взаимодействии.

`ServerRC2Key` - Ключ сервера

`ClientId`, `TerminalId` - идентификаторы устройства в системе мониторинга. Для устройства паркомат должны совпадать с параметром `TheDeviceNumber`.

`SoftwareVersion` - версия установленного программного обеспечения.

Предполагается что этот параметр устанавливается скриптом обновления ПО успешному результату обновления. А ПО паркомата использует данный параметр при отправке запроса на сервер мониторинга для уточнения своей версии.

`SendPeriodMaxMsec`, `SendPeriodMinMsec` - максимальная и минимальная периодичность отправки запросов в мониторинг. Очередь сообщений адаптивная и зависит от ее длины и приоритетов. Данные параметры задают пороги по которым будет отправляться соответственно самая короткая очередь и самая длинная.

`CheckTimeoutInTechSec` - время на установление соединения с мониторингом при просыпании в техническом режиме

`SendingRetriesTimeoutSec` - Время в секундах на попытки отправить сообщение в случае отсутствия связи, после чего считаем, что связи нет.

Конфигурационные параметры зависимостей устройств секция `DevicesTree`.

Как указывалось ранее секция `DevicesTree` обязательна. По ней `DevicesManager` определяет какие устройства должны быть загружены и какие зависимости между ними устанавливаются.

Формат секции состоит перечня устройств, в которые вложены необходимые для устройства зависимости и в зависимости вложены соответствующие устройства. Пример записи следующий:

```
<DevicesTree>
  <Device Name="TheDevice">
    <Depends Type="all">
      <Device Name="Printer"/>
      <Device Name="BankCardTerminal"/>
    </Depends>
    <Depends Type="one"/>
    <Depends Type="optional">
      <Device Name="ServiceButton"/>
      <Device Name="Keyboard"/>
    </Depends>
  </Device>
</DevicesTree>
```

Данный пример говорит о том что в ПО будут загружены устройства `TheDevice`, `Printer`, `BankCardTerminal`, `Keyboard`, `ServiceButton`. Устройство `TheDevice` зависит от устройств `Printer` и `BankCardTerminal` и опционально от `Keyboard`.

Зависимость типа `all` – это обязательная зависимость с типом “И” говорит о том, что устройство готово к обслуживанию (успешный статус), если все из устройств, перечисленных в зависимости готовы к обслуживанию. И если хоть одно из устройств перечисленных в зависимости не готово к обслуживанию (в ошибке), то и зависимое устройство не готово к обслуживанию.





Зависимость типа `one` – это обязательная зависимость с типом “ИЛИ” говорит о том, что устройство готово к обслуживанию (успешный статус), если все хоть одно из устройств, перечисленных в зависимости готовы к обслуживанию. И если все из устройств перечисленных в зависимости не готово к обслуживанию (в ошибке), то и зависимое устройство не готово к обслуживанию.




Зависимость типа `optional` – это не обязательная (опциональная) зависимость. Такая зависимость никак не влияет на возможность обслуживания пользователя, но требуется для некоторых алгоритмов работы устройств например алгоритм `TheDevice` ищет в перечне зависимости `ServiceButton` и проверяет ее состояние по которому вызывает сервисное меню и если такой зависимости не будет, то вызов меню не будет осуществляться.

2.2. Описание клавиш клавиатуры

//перекодировщик входящих кодов от клавиатуры (0..47) в системные клавиши

Кнопка клавиатуры	название	kmap	ASCII	
0	zero	keycode 11	48	
1	one	keycode 2	49	
2	two	keycode 3	50	
3	three	keycode 4	51	
4	four	keycode 5	52	
5	five	keycode 6	53	
6	six	keycode 7	54	
7	seven	keycode 8	55	
8	eight	keycode 9	56	
9	nine	keycode 10	57	
A	A	keycode 30	65	
B	B	keycode 48	66	
C	C	keycode 46	67	
D	D	keycode 32	68	

E	E	keycode 18	69	
F	F	keycode 33	70	
G	G	keycode 34	71	
H	H	keycode 35	72	
I	I	keycode 23	73	
J	J	keycode 36	74	
K	K	keycode 37	75	
L	L	keycode 38	76	
M	M	keycode 50	77	
N	N	keycode 49	78	
O	O	keycode 24	79	
P	P	keycode 25	80	
Q	Q	keycode 16	81	
R	R	keycode 19	82	
S	S	keycode 31	83	
T	T	keycode 20	84	
U	U	keycode 22	85	
V	V	keycode 47	86	
W	W	keycode 17	87	
X	X	keycode 45	88	
Y	Y	keycode 21	89	
Z	Z	keycode 44	90	
	Транспортное средство, запятая (сomma « , »)	keycode 125	44	
	минус (minus « - »)	keycode 12	12	
	плюс (plus « + »)	keycode 78	78	
	знак больше (greater « > »)	keycode 126	62	

		ВВОД (Enter, Return)	keycode 28		
		ВЫХОД, отмена, escape (Esc) Return			
		Backspace	keycode 14		
		собака (@), коммерческая at	keycode 26	64	
Режим Mode		Звездочка, asterisk (*)	keycode 55	42	
 Старт Start		кнопка включения			