Прогаммное обеспечение PayCommGate
Редакция PayComm.DeviceManagerSerivce

Инструкция по эксплуатации
Версия 1.0

ООО «Пэйкомм». Москва 2018.

Оглавление

1. Общие сведения

Программное обеспечение предназначено для работы в устройствах самообслуживания, позволяет управлять устройствами приема денежных средств и выдачи денежных средств. Управление устройствами реализовано в виде Веб-сервиса.

Список поддерживаемого оборудования:

- Купюроприемники, использующие протокол CCNET (CashCode MFL, CashCode FL)
- Монетоприемники, использующие протокол CCTALK (NRI G13)
- Хоппер - Smart Hopper
- Диспенсеры – Puloon LCDM 2000, Puloon LCDM 4000

Операционная система:

- Windows XP Embedded SP3;
- Windows XP Professional SP3;
- Windows 7 и выше (Windows 8, Windows 10)

Пререквизиты (пререквизиты - дополнительные компоненты, необходимые для функционирования программного обеспечения):

- Microsoft Framework .Net 4.0

Протокол управления устройствами реализован по http посредством Get запросов.

2. Установка, настройка и запуск программного обеспечения.

ПО может поставляться в виде инсталлятора либо в виде архива. Для установки ПО из архива, рекомендуем содержимое архива распаковать на диск c:\ в корень. В архиве папка Client. Запускающий файл необходимо поместить в автозагрузку либо прописать в реестре Windows, где запускаются программы при старте операционной системы, либо запускать ПО из сторонней программы (Графического интерфейса, например). Программа сама не прописывается в автозагрузку, сделано специально, чтобы была возможность чтобы можно было настроить под свои нужды.

Запуск программы происходит файлом PayComm.DeviceManager.exe. При запуске программы отражается окно (Рисунок 1) с процессом автоматического поиска оборудования. Как только поиск оборудования будет закончен, окно скроется в Tray (Рисунок 2). Запуститься Веб-сервис управления устройствами.

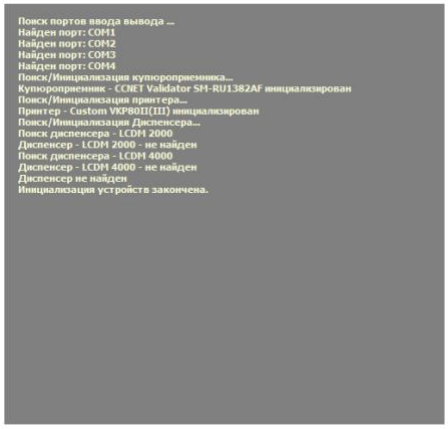

Рисунок 1 - Внешний вид Log-окна программного обеспечения

Также в данное окно пишется действия работы некоторых устройств, например, в купюроприемник внесли купюру.

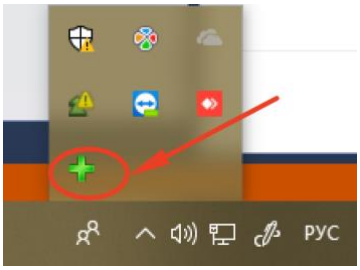

Рисунок 2 - Внешний вид Tray

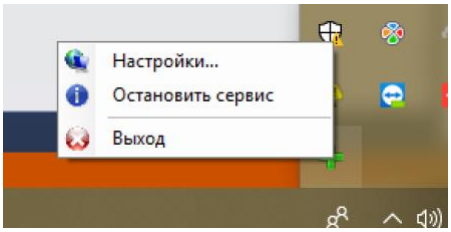

Рисунок 3 - Внешний вид меню приложения

Вызвать меню можно нажатием правой кнопки мыши (Рисунок 3). В меню пункт «Настройки» отразит Log окно (Рисунок 1) (Лог окно можно просто закрыть, оно исчезнет в Tray), Log – окно можно также открыть двойным щелчком мыши по Tray иконки и таким же образом закрыть, пункт «Остановить сервис» - остановит Веб-сервис управления устройствами и пункт меню поменяет название на «Запустить Сервис» (по нажатию, которого Веб-сервис обратно запустится), пункт меню «Выход» полностью выгрузит программу.

3. Протокол взаимодействия с устройствами.

Веб-сервис слушает порт 8888. Для обеспечения безопасности рекомендуется закрыть файерволом порт 8888, чтобы невозможно было обратиться к сервису из внешней сети. Веб-сервис предназначен для работы в локальной среде и служит связующим звеном между Графическим интерфейсом обслуживания клиента и уровнем устройств.

3.1. Общий принцип работы запросов и ответов.

Все запросы необходимо выполнять по ссылке http://localhost:8888/devices .
Запросы к Веб-сервису осуществляются по протоколу HTTP методом GET.
Формат запроса выглядит следующим образом:

http://localhost:8888/devices?method=<имя_метода>&<имя_параметра1>=<значение_параметра1>&<имя_параметра2>=<значение_параметра2>.....&<имя_параметраN>=<значение_параметраN>&<имя_параметраN+1>=<значение_параметраN+1>

Если метод не имеет параметров, то:

http://localhost:8888/devices?method=<имя_метода>

Ответ на запрос возвращается в формате xml.
Ответ всегда состоит из базовой структуры xml ответа:

```
<Response Result="1" Error="0" Message="Ok">
   .....
</Response>
```

Где, Result – всегда равен 1 если запрос сделан корректно, по протоколу, 0 – если метод не существует. Error – Код ошибки, Message – описание ошибки, либо технологическое сообщение.

Таблица 1 - коды ошибок в ответе

Код ошибки	Описание
1	Происходит при открытии сессии или закрытии. 1) Если сессия не закрыта и отправлен запрос на открытие сессии повторно. 2) Если отправлен запрос на закрытие сессии, а открытой сессии нет.
400	Метод, указанный в запросе, не существует в протоколе.
401	Не указан метод в запросе.
500	При обращении к устройствам в момент инициализации ПО (т.е. ПО еще не до конца загрузилось)
501	Устройство не найдено
502	Устройство занято процессом
503	Устройство в состоянии ошибки

Структура ответа xml в зависимости от запроса будет пополнятся различными тэгами со значениями. Различные структуры будут рассмотрены ниже в следующих пунктах протокола.

3.2. Методы, используемые в Веб-сервисе.

Список команд и краткое описание отражены в таблице ниже (Таблица 2).

Таблица 2 - Список методов веб-сервиса

Методы	Краткое описание
session_open	Открыть сессию
session_close	Закрыть сессию
get_status	Получить полный статус
get_status_validator	Получить статус купюроприемника
get_status_dispenser	Получить статус диспенсера
get_status_coinvalidator	Получить статус монетоприемника
get_status_hopper	Получить статус хоппера
coinvalidator_start	Включить прием монет в монетоприемнике
coinvalidator_stop	Выключить прием монет в монетоприемнике
validator_start	Включить прием купюр в купюроприемнике
validator_stop	Выключить прием купюр в купюроприемнике
dispenser_present	Выдать купюры из диспенсера
hopper_present	Выдать монеты из хоппера

3.2.1. Метод «session_open». Сессия и описание структура сессии в xml ответах.

Сессия — это объект временного хранения данных, полученных от устройств за время обслуживания клиента. Сессия хранит данные о принятых купюрах, монетах, выданных монет, выданных купюр.
Как правило, сессию необходимо открывать, когда наступил момент работы с устройствами (принять или выдать денежные средства).

Во всех запросах статуса, если сессия была открыта, будет в ответе возвращена структура сессии в формате xml. Структура меняется в зависимости от совершенных действий, если были приняты купюры – появляются тэги с информацией о принятых купюрах, были приняты монеты - появляются тэги с информацией о принятых монетах, была выдана сдача из диспенсера – появится информация о кол-ве выданных купюр из каких кассет, а также из каких кассет кол-во купюр отбракованных и сброшенных в reject лоток диспенсера, были выданы монеты – появится информация о выданных монетах.

Если в процессе обслуживания были внесены и монеты, и купюры, а также выдана сдача и купюрами, и монетами, то в данном случае можно наблюдать в ответе на статусы полную структуру сессии. А если, например, были внесены средства купюрами и выдана сдача монетами то в структуре будет отражаться только информация о внесенных купюрах и выданной сдачи монетами.

Полный формат структуры сессии:

```
<Session SessionId="00006367120084642600" SessionIdCreateTime="2018-08-30T04:40:46.4264613+03:00" TotalAmount="10208">
    <StackedBanknotes>     - Блок принятых купюр за сессию
      <Banknote Nominal="5000" StackedTime="30.08.2018 5:06:57" />  - каждая принятая банкнота записывается в блок, имеет параметр
      <Banknote Nominal="200"  StackedTime="30.08.2018 5:07:03" />     StackedTime время принятия банкноты. Такой Лог применим при
      <Banknote Nominal="5000" StackedTime="30.08.2018 5:07:10" />     разборе нештатных или спорных ситуациях. Также обозначен номинал -
    </StackedBanknotes>                                                - Nominal
    <StackedCoins>     - Блок принятых монет за сессию
      <Coin Nominal="1" StackedTime="30.08.2018 5:06:57" />    - каждая принятая монета записывается в блок, имеет параметр
      <Coin Nominal="2" StackedTime="30.08.2018 5:07:03" />      StackedTime время принятия банкноты. Также обозначен номинал - Nominal.
      <Coin Nominal="5" StackedTime="30.08.2018 5:07:10" />
    </StackedCoins>
    <DispenserCassettes> - Блок кассет диспенсера. Отражает какое кол-во купюр в каждой кассете было выдано за сессию.
      <Cassette Id=”1” DispensedCount=”3” RejectedCount=”0”>     Id – номер кассеты, DispensedCount – кол-во выданных банкнот, RejectedCount -
      <Cassette Id=”2” DispensedCount=”1” RejectedCount=”1”>     - кол-во отбракованных банкнот и сброшенных в лоток для отбракованных купюр -
       .                                                          RejectTray диспенсера
       .
       .
      <Cassette Id=”n”   DispensedCount=”0” RejectedCount=”0”>
      <Cassette Id=”n+1” DispensedCount=”1” RejectedCount=”0”>
    </DispenserCassettes>
    <DispensedCoins> - Блок выданных монет. Отражает какое кол-во монет каждого номинала было выдано за сессию. Id – номинал, DispensedCount – кол-во
      <Coin Id=”1” DispensedCount=”3” > выданных монет.
      <Coin Id=”2” DispensedCount=”1” >
      <Coin Id=”3” DispensedCount=”0” >
       .
       .
       .
      <Coin Id=”n”   DispensedCount=”0” >
      <Coin Id=”n+1” DispensedCount=”0” >
    </DispensedCoins>
</Session>
```

SessionId – уникальный номер сессии, SessionIdCreateTime – время создания сессии, TotalAmount – Сумма внесенных денежных средств в рублях за сессию (сумма всех принятых банкнот и монет в денежном эквиваленте, например, в данном случае в рублях).

Запрос на открытие сессии «session_open»:
http://localhost:8888/devices?method=session_open

Положительный ответ может быть получен, если предыдущая сессия была закрыта.

```
<Response Result="1" Error="0" Message="Ok">
  <Session SessionId="00006367120084642600" SessionIdCreateTime="2018-08-30T04:40:46.4264613+03:00" TotalAmount="0" />
</Response>
```

Ошибочный ответ может быть получен, если текущая сессия не была закрыта.

```
<Response Result="1" Error="1" Message="Current Session not closed." />
```

3.2.2. Метод «session_close»

Данный метод закрывает сессию. В случае положительного ответа выдает полную, либо частичную структуру сессии, описанной в пункте 3.2.1

Запрос на закрытие сессии «session_close»:
http://localhost:8888/devices?method=session_close

Пример положительного ответа на запрос:

```
<Response Result="1" Error="0" Message="Ok">
<Session SessionId="00006367120084642600" SessionIdCreateTime="2018-08-30T04:40:46.4264613+03:00" TotalAmount="10200">
    <StackedBanknotes
      <Banknote Nominal="5000" StackedTime="30.08.2018 5:06:57" />
      <Banknote Nominal="200"  StackedTime="30.08.2018 5:07:03" />
      <Banknote Nominal="5000" StackedTime="30.08.2018 5:07:10" />
    </StackedBanknotes>
 </Session>
</Response>
```

Ошибочный ответ может быть получен, если нет открытой сессии:

```
<Response Result="1" Error="1" Message="Opened Session not found." />
```

3.2.3. Метод «get_status», «get_status_<deviceType>»

Данные методы возвращают статусы работы (поведения) устройства. «get_status» - возвращает статусы всех устройств, остальные только конкретного устройства. Следует обратить внимание, что при вызове статуса, обязательно, если открыта сессия, то также в статусе будет возвращена структура сессии полная или частичная (описано в пункте 3.2.1), либо вовсе может не быть если сессия не открыта.

Пример запроса полного статуса «get_status»:
http://localhost:8888/devices?method=get_status

Пример ответа:

```
<Response Result="1" Error="0" Message="Ok">
  <Session SessionId="00006367120659724200" SessionIdCreateTime="2018-08-30T06:16:37.2424881+03:00" TotalAmount="0" />
  <Validator Status="0" IsStarted="False" IsError="False" IsBusy="False" Firmware="SM-RU1382AF" Port="COM4" />
  <Dispenser Status="200" IsError="true" IsBusy="false" Firmware="" Port="NA" />
  <CoinValidator Status="200" IsStarted="False" IsError="true" IsBusy="false"  Firmware="" Port="NA" />
  <Hopper Status="200" IsError="true" IsBusy="false" Firmware="" Port="NA" />
</Response>
```

Пример запроса статуса «get_status_validator»:
http://localhost:8888/devices?method=get_status_validator

Пример ответа:

```
<Response Result="1" Error="0" Message="Ok">
  <Session SessionId="00006367120659724200" SessionIdCreateTime="2018-08-30T06:16:37.2424881+03:00" TotalAmount="0" />
  <Validator Status="0" IsStarted="False" IsError="False" IsBusy="False" Firmware="SM-RU1382AF" Port="COM4" />
</Response>
```

Примеры запроса статуса для других устройств:

http://localhost:8888/devices?method=get_status_dispenser
http://localhost:8888/devices?method=get_status_coinvalidator
http://localhost:8888/devices?method=get_status_hopper

Все устройства имеют параметры:

Status – Код состояния устройства. 0 – ошибок нет, >=200 – ошибочное состояние устройства при котором устройство не может функционировать.
IsError – логическое (bool) значение находится ли устройство в состоянии ошибки
IsStarted – логическое (bool) значение только для устройств приема денежных средств, показывает включено ли устройство на прием денежных средств.
IsBusy - логическое (bool) значение показывает, что устройство занято процессом либо принятия купюры/монеты, либо выдачи сдачи. Данное свойство необходимо использовать в графическом интерфейсе процесса обслуживания, особенно при принятии денежных средств, обычно на экране где отражается прием денежных средств располагают кнопки «Оплатить» , «Отмена» или «Назад» (только в случае если ни одной денежной единицы не было внесено отражают кнопки отмена или оплатить). Если IsBusy = true, то необходимо все кнопки с экрана скрывать. Далее алгоритм обслуживания, как правило, если были внесены любые денежные средства отражают только кнопку «оплатить» в том случае если все устройства приема денежных средств имеют значение IsBusy = false. Но если не было внесено ни одной денежной единицы, то следует отобразить кнопки «Отмена» и «Назад» а кнопку «Оплатить» скрыть. Для реализации данного алгоритма необходимо делать циклично запросы статуса устройств приема денежных средств с промежутком времени в 300 миллисекунд на окне внесения средств.
Firmware – информация о версии микропрограммы (прошивки) устройства, если она присутствует.
Port – информация о том, к какому порту подключено устройство.

3.2.4. Методы «coinvalidator_start», «coinvalidator_stop>» , «validator_stop>», «validator_start>»

Метод coinvalidator_start включает монетоприемник на прием монет.
Метод validator_start включает купюроприемник на прием купюр.
Методы _stop выключают прием денежных средств.

Примеры запросов:

http://localhost:8888/devices?method=coinvalidator_start
http://localhost:8888/devices?method=coinvalidator_stop
http://localhost:8888/devices?method=validator_start
http://localhost:8888/devices?method=validator_stop

Все запросы дают положительный ответ если устройства присутствуют в системе и не имеют ошибочного статуса.

```
<Response Result="1" Error="0" Message="Ok" />
```

В случае ошибки:

```
<Response Result="1" Error="Код ошибки" Message="Описание ошибки" />
```

3.2.5. Методы «dispenser_present», «hopper_present>»

Данные методы выдают денежные средства у этих методов есть дополнительные параметры.

dispenser_present – выдает купюры из кассет диспенсера Puloon LCDM2000 или LCDM4000 (4 кассетный)

hopper_present – выдает монеты из устройства Smart Hopper

примеры запросов:

если LCDM 2000
(если из какой-то кассеты не надо выдавать то значение ставим равное 0)
http://localhost:8888/devices?method=dispenser_present&count1=1&count2=5

если LCDM 4000
(если из какой-то кассеты не надо выдавать то значение ставим равное 0)
http://localhost:8888/devices?method=dispenser_present&count1=1&count2=0&count3=0&count4=2

Формат запроса для хоппера немного отличается в параметрах
http://localhost:8888/devices?method=hoper_present&nominals=1-3;2-5;5-3

здесь указываем только те номиналы и кол-во которые надо выдать в отличие от кассет диспенсера
nominals=<номинал_монеты1>-<кол-во>;<номинал_монеты2>-<кол-во>.....<номинал_монетыN>-<кол-во>

Все запросы дают положительный ответ если устройства присутствуют в системе и не имеют ошибочного статуса.

```
<Response Result="1" Error="0" Message="Ok" />
```

В случае ошибки:

```
<Response Result="1" Error="Код ошибки" Message="Описание ошибки" />
```

Далее необходимо циклично проверять статус устройства на параметр IsBusy как только он поменяет значение на false можно считать в сессии информацию о том сколько по факту было выдано кол-во купюр из каждой кассеты. К сожалению, даже если рассчитать точное кол-купюр заправленных в кассетах может произойти отбраковка купюр в лоток RejectTray и исходя из этого запрошенное кол-во купюр из кассет может быть меньше если купюры в кассете заканчиваются. Поэтому всегда надо сравнивать кол-во выданных купюр по факту.