

AtlasFPNet.exe ver. 2.00

Руководство по эксплуатации

(Инсталляция, администрирование, тестирование и запуск)

Содержание

1. Введение	3
2. Назначение ПО	4
3. Комплект поставки	5
4. Установка ПО	6
5. Тестирование объекта и регистратора с помощью внутреннего теста	8
6. Тестирование объекта и регистратора с помощью внешнего теста	14
7. Образцы для обработок в 1С 8.	16
8. Конфигурирование ПО	26
9. Особенности запуска ПО	29
10. Приложение. Дополнительные команды для работы с протоколом Криптон вер. 5.03.	31

1. Введение

В данном документе описаны особенности инсталляции, администрирования, тестирования и запуска COM-объекта **AtlasFPNet.exe** для фискальных регистраторов ДП "Компании "ATLAS" (**SystemGroup** Украина), подключенных с помощью Ethernet-подключений в локальную сеть, а также в более широкие сети с помощью более мощного сетевого оборудования.

Для проведения инсталляции, администрирования и тестирования необходимо владеть базовыми знаниями и навыками выполнения этих операций в среде Windows. Перед чтением данного документа необходимо изучить документы относящиеся к описанию фискальных регистраторов, которые подключаются в сеть.

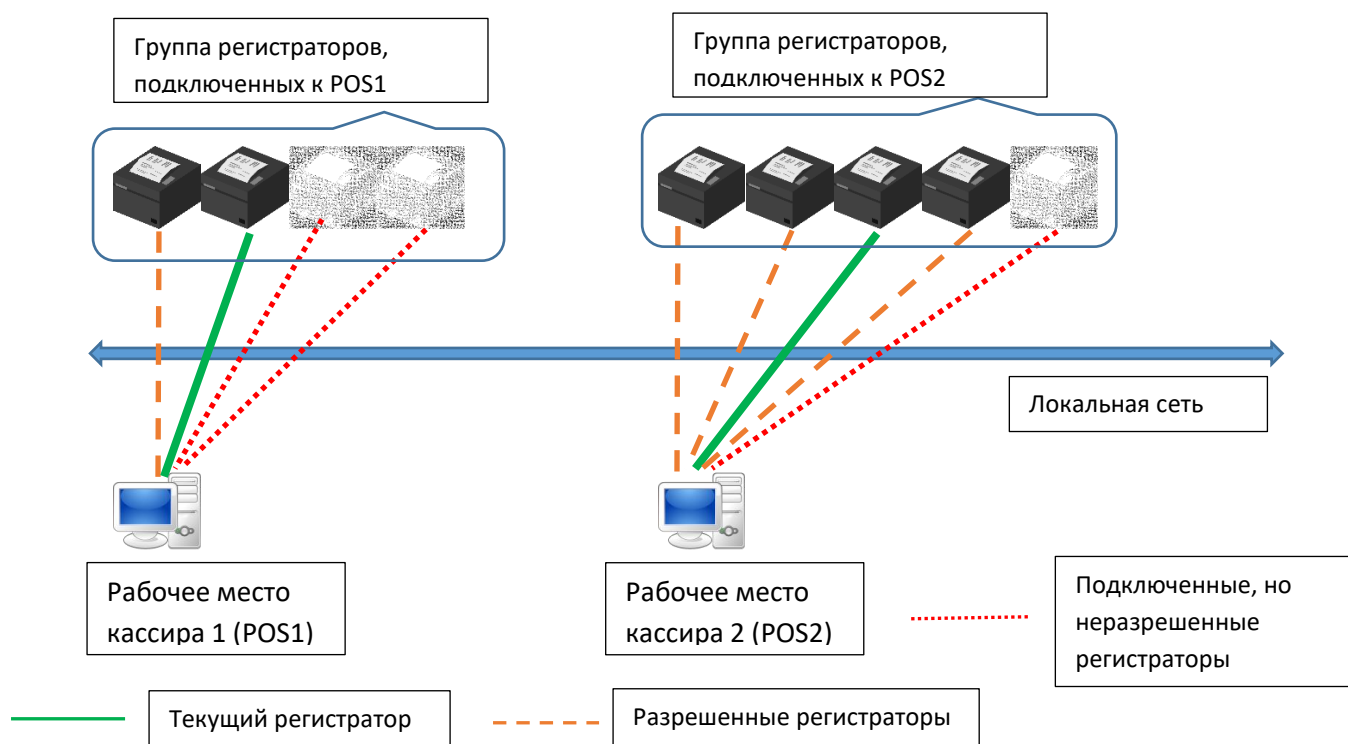
По вопросам функционирования, сопровождения и исправления ошибок в COM-объекте обращайтесь по E-mail: Vladimir.Afanasiev@systemgroup.com.ua.

2. Назначение ПО

Данное ПО предназначено для выполнения торговых операций на фискальных регистраторах, подключенных в сеть с помощью Ethernet-соединения. При этом предполагается, что рабочие места кассиров (операторов) находятся в той же сети, что и регистраторы. **AtlasFPNet.exe** – представляет собой специальную коммуникационную библиотеку (выполненную как COM-объект), к которой обращается ПО рабочего места кассира для передачи и получения данных к одному из регистраторов, расположенных в сети. Коммуникация выполняется на базе протокола TCP/IP, при этом COM-объект играет роль TCP/IP сервера, а регистратор – TCP/IP клиента. Для того, чтобы такое соединение было эффективно необходимо перед началом работы выполнить программирование соответствующих параметров регистратора, указав в них, в частности, TCP/IP адрес сервера, то есть TCP/IP адрес компьютера, на котором установлен сервер **AtlasFPNet.exe**.

В процессе работы регистраторы подключаются к “своим” COM-объектам, то есть каждый к тому, к которому его запрограммировали. Вполне нормальная ситуация, когда к одному COM-объекту подключаются два и более регистраторов. Чтобы упорядочить эту ситуацию, вводятся понятия “текущего регистратора” и “разрешенных регистраторов”.

Фактически получается, что к одному объекту может подключиться группа регистраторов, но у каждого объекта есть перечень “разрешенных регистраторов” и только с ними разрешено работать данному объекту. В то же время, предполагается, что в каждый момент времени COM-объект может работать только с одним из регистраторов (тем самым, который назначен “текущим регистратором” программным обеспечением рабочего места кассира). Однако, вполне допускается, что по ходу работы кассир может переподключаться то к одному, то к другому из уже подключенных к COM-объекту “разрешенных регистраторов”, назначая такой регистратор “текущим регистратором”.



3. Комплект поставки

Комплект поставки состоит:

- 1) COM объект **AtlasFPNet.exe**
- 2) Внутренний тест (включен в состав COM объекта)
- 3) Внешний тест **ExtClient.exe** (с исходным кодом)
- 4) Обработка для 1С 8.2 **SystemGroup_ATLAS_v2_Net.epf**
- 5) Руководство программиста
- 6) Руководство по эксплуатации (данное руководство)

4. Установка ПО

Все COM объекты, установленные на компьютере, делятся на два больших класса:

- 1) **Dll** или внутренние серверы – InprocServers
и
- 2) **Exe** или локальные серверы – LocalServers

Таким образом, **AtlasFPNet.exe** относится к локальным серверам. Это означает, что для установки объекта его (и файл AtlasFPNet.ini) надо записать в каталог откуда он будет в дальнейшем стартовать. После этого надо запустить файл **AtlasFPNet.exe** на выполнение (правая кнопка мышки + Запуск от имени администратора) и затем закрыть (завершить выполнение) объекта нажатием соответствующей кнопки на форме. Собственно, на этом и заканчивается этап установки объекта. Обращаю внимание, что файл AtlasFPNet.ini является важным атрибутом объекта поскольку предназначен для дальнейшего администрирования объекта в процессе его жизненного цикла.

Вот так будет выглядеть окно объекта после установочного запуска (при условии, что файл AtlasFPNet.ini не будет предварительно корректироваться).

А так выглядит файл AtlasFPNet.ini:

[Net]

Port=34952

[FiscalPrinters]

IDDEV1=402400076

IDDEV2=402300005

IDDEV3=402300002

IDDEV4=402300004

IDDEV5=402300009

IDDEV6=402300011

IDDEV7=402300023

IDDEV8=402300111

[Start]

;Program=Quiet

;Program=Open

Program=Test

[QuietNotification]

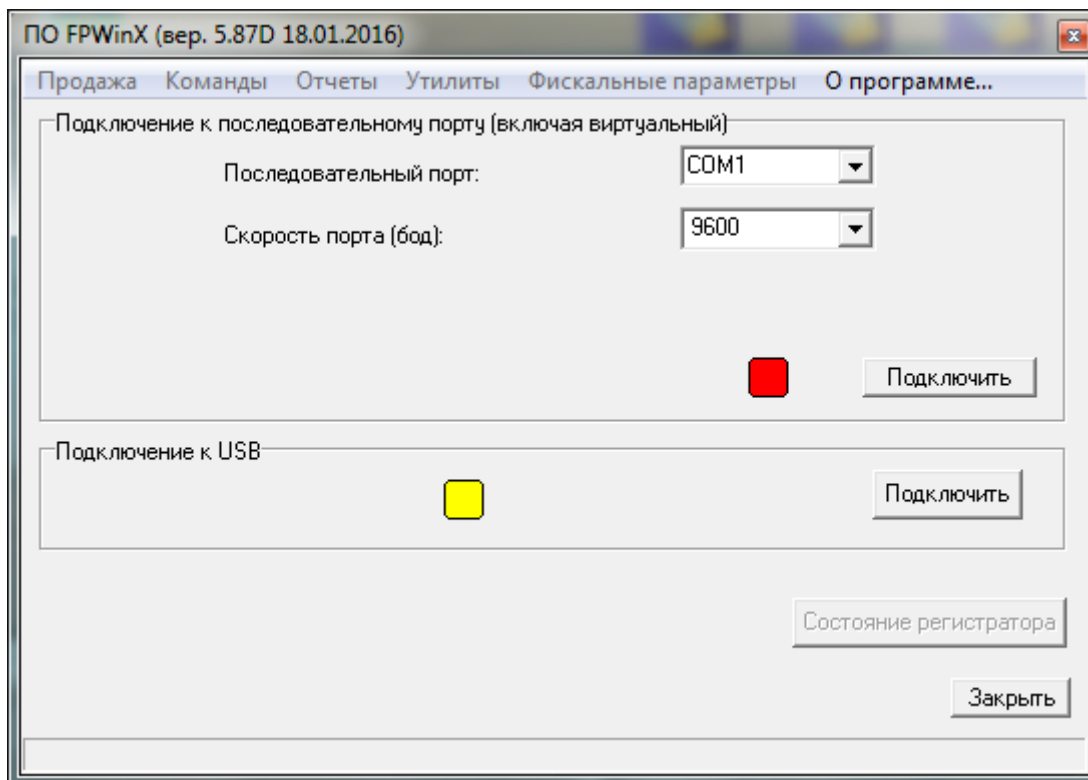
Connected=Yes

Disconnected=Yes

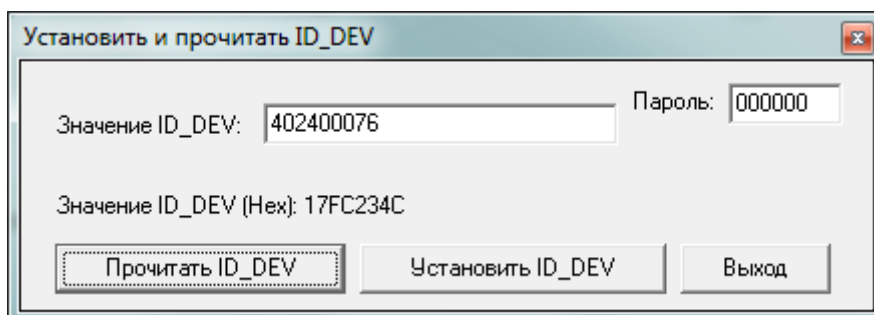
Error=Yes

5. Тестирование объекта и регистратора с помощью внутреннего теста

Прежде чем тестировать регистратор с помощью внутреннего теста объекта, необходимо настроить сетевые параметры регистратора. Для этих целей используется ПО FPWinX, к которому через последовательный порт подключен регистратор:



Прежде всего уточните ID_DEV Вашего регистратора. Для этого в утилите FPWinX выберите меню **Фискальные параметры** подменю **Установить и прочитать ID_DEV**, после этого нажмите кнопку **Прочитать**:



При настройке регистратора необходимо установить следующие параметры на регистраторе:

- Сетевой адрес (TCP/IP) регистратора;
- TCP/IP порт для работы с COM объектом;
- Маску сети;

- Сетевой адрес компьютера, на котором находится целевой COM объект;
- А так же имена кассиров

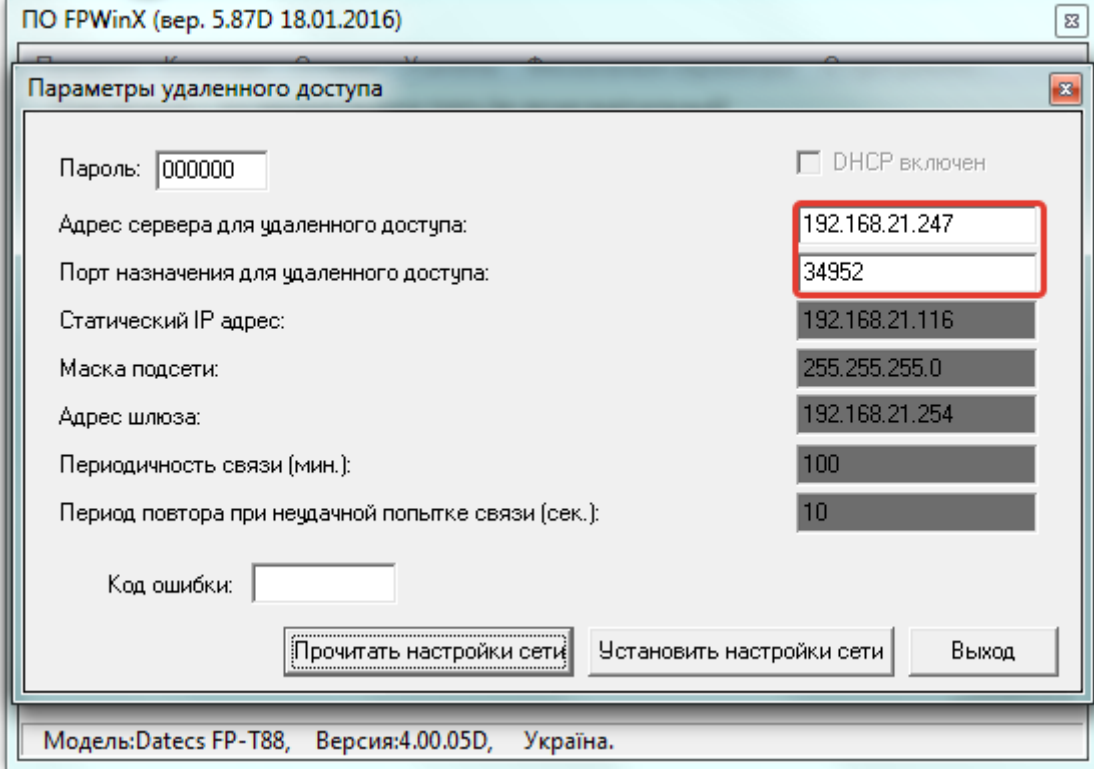
Может оказаться, что некоторые параметры (например, адрес регистратора и адрес подсети) уже были настроены предварительно. Тогда их можно по желанию перенастроить или нет, важно, чтобы это было сделано корректно, с учетом реальной конфигурации локальной сети. Для настройки сетевого адреса регистратора и маски подсети зайдите в ПО FPWinX, подключите регистратор и зайдите в меню **Фискальные параметры**, подменю **Параметры связи с эквайером**.

После этого надо нажать кнопку **Прочитать настройки сети**. В управляющих элементах появятся текущие настройки, как указывалось выше они могут быть изменены, чтобы обеспечить соответствие реалиям локальной сети:

Примечание. После выполнения программирования параметров связи эквайера (кнопка **Установить настройки сети**) регистратор выполнит перезагрузку

Теперь можно установить TCP/IP порт для работы с объектом и сетевой адрес компьютера, на котором находится целевой COM объект, необходимо закрыть форму

Параметры связи и зайти в меню **Фискальные параметры**, подменю **Параметры сетевого доступа**:



ПО FPWinX (вер. 5.87D 18.01.2016)

Параметры удаленного доступа

Пароль: 000000 ☐ DHCP включен

Адрес сервера для удаленного доступа: 192.168.21.247

Порт назначения для удаленного доступа: 34952

Статический IP адрес: 192.168.21.116

Маска подсети: 255.255.255.0

Адрес шлюза: 192.168.21.254

Периодичность связи (мин.): 100

Период повтора при неудачной попытке связи (сек.): 10

Код ошибки:

Модель: Datecs FP-T88, Версия: 4.00.05D, Украина.

На вышеприведенном рисунке выделены управляющие поля, соответствующие устанавливаемым параметрам.

Для создания необходимых кассиров надо использовать FPWinX – зайдите в меню **Утилиты**, подменю **Параметры операторов**.

Прочитайте параметры операторов с номерами 0 и 1. Для этого в окне **Номер оператора** укажите 0(1) и нажмите кнопку **Прочитать**.

Если окажется, что какой-либо из этих операторов не запрограммирован, то обязательно установите его до начала работы с объектом.

The dialog box is titled "Параметры операторов". It contains several input fields and buttons:

- "Номер оператора:" with a text box containing "0".
- A section titled "Установить параметры оператора" containing:
 - "Имя:" with a text box containing "Петренко".
 - "Пароль оператора:" with a text box containing "000000" and a "Установить" button.
- A section titled "Прочитать параметры оператора" containing:
 - "Имя:" with a text box containing "Петренко".
 - A checked checkbox labeled "Выполнялись продажи" and a "Прочитать" button.
- A "Удалить оператора" button at the bottom center.
- A "Пароль:" text box containing "000000" and a "Закреть" button at the bottom right.

После этих настроек можно выполнить перевод регистратора в режим работы через Ethernet. Для этого надо с помощью программы FPWinX зайти в меню **Утилиты** подменю **Служебные счетчики**.

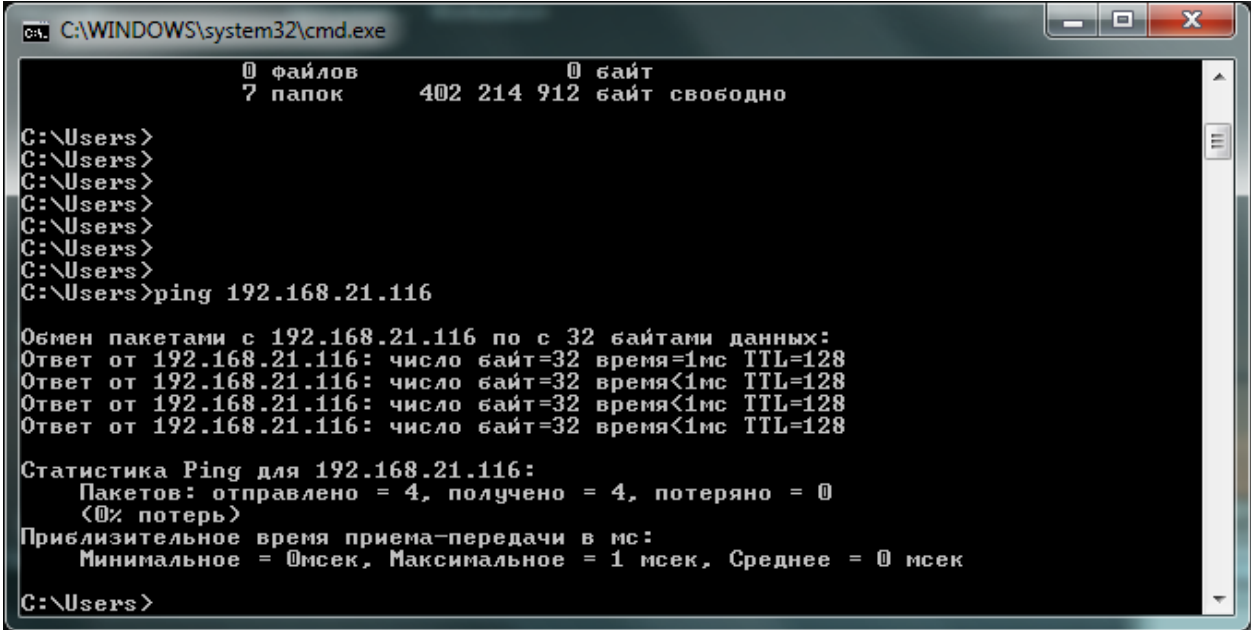
В падающем списке необходимо выбрать счетчик Выбор режима связи и установить там значение 4. Регистратор переходит в режим работы через Ethernet.

The dialog box is titled "Служебные счетчики". It contains the following elements:

- A section titled "Чтение счетчиков" containing:
 - "Служебный счетчик:" with a dropdown menu showing "Выбор режима связи".
 - "Значение:" with a text box containing "4".
 - A "Установить" button (highlighted with a red rectangle) and a "Прочитать" button.
- A "Пароль:" text box containing "000000" and a "Закреть" button at the bottom right.

Здесь мы не рассматриваем случаи подключения ПО в более сложную конфигурацию сети, когда надо специальным образом настраивать маршрутизаторы и прокидывать TCP/IP порты (это работа для опытных администраторов сети). Однако, для нашей, более простой конфигурации уже наступило время выполнить первые тесты, которые могут

подтвердить корректность сетевых настроек и наличие доступа к регистратору и компьютеру. Для этих целей используется обычный ping регистратора через сеть. Вот обычный ping корректно настроенного регистратора:



```
C:\WINDOWS\system32\cmd.exe
0 файлов          0 байт
7 папок          402 214 912 байт свободно

C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>
C:\Users>ping 192.168.21.116

Обмен пакетами с 192.168.21.116 по 32 байтами данных:
Ответ от 192.168.21.116: число байт=32 время=1мс TTL=128
Ответ от 192.168.21.116: число байт=32 время<1мс TTL=128
Ответ от 192.168.21.116: число байт=32 время<1мс TTL=128
Ответ от 192.168.21.116: число байт=32 время<1мс TTL=128

Статистика Ping для 192.168.21.116:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 1 мсек, Среднее = 0 мсек

C:\Users>
```

Так же рекомендуется пропинговать компьютер, на котором установлен COM объект.

Теперь можно перейти к тестированию с помощью внутреннего теста COM объекта. Прежде всего надо открыть файл AtlasFPNet.ini, который должен находиться в том же каталоге, что и COM объект и настроить его.

Так следующие строчки определяют, что при следующем запуске COM объекта будет запущен внутренний тест:

```
[Start]
;Program=Quiet
;Program=Open
Program=Test
```

Здесь незакомментированной является только строчка **Program=Test**, которая и обозначает, что при запуске COM объекта запустится внутренний тест. Две другие строчки (закомментированные) используются если необходимо запустить не тест, а нормальное выполнение объекта.

Кроме того, в раздел **[FiscalPrinters]** надо включить IDDEV подключаемого регистратора – этот раздел представляет собой список разрешённых регистраторов. Например, если надо разрешить работу с двумя регистраторами - IDDEV=402400076 и IDDEV=402401111, то это должно выглядеть так:

[FiscalPrinters]

IDDEV1=402400076

IDDEV2=402301111

Теперь, когда все настроено, можно запускать внутренний тест. Для этого надо двойным кликом левой кнопки мышки запустить объект и если все настроено правильно, то на экране появится такая форма:

Здесь на рисунке выше видно, что регистратор с IDDEV равным 402400076 уже подключился к COM объекту и “виден” внутреннему тесту. На самом деле процесс подключения конкретного регистратора существенно асинхронен и с момента включения регистратора и запуска COM объекта может пройти некоторое время пока регистратор как клиент подключится к COM объекту как к серверу. Если бы в сети были включены несколько разрешенных для данного COM сервера регистраторов, то они при нормально работающей сети появились бы в том же списке (разумеется асинхронно по времени).

Кнопка **Подключиться к объекту** и выбор одного из подключенных регистраторов позволит выбирать те или действия с регистраторами, выполняя их через локальную сеть. По ходу выполнения теста можно переключаться от одного регистратора к другому.

Примечание. Выбирая ту или иную из предложенных операций необходимо соблюдать осторожность, чтобы выполнить каких-либо невозвратных действий на регистраторе (например, Z-отчет) в случае если он фискализирован.

6. Тестирование объекта и регистратора с помощью внешнего теста

Тестирование объекта внешним тестом позволяет выполнять тестирование в условиях близких к работе пользовательских приложений, которые тоже работают как внешние по отношению к COM объекту.

Перед запуском внешнего теста необходимо установить в файле AtlasFPNet.ini параметр Program или так:

[Start]

Program=Quiet

Или так:

[Start]

Program=Open

При этом запускать COM объект напрямую (как это делается в случае внутреннего теста) не следует, COM объект будет подниматься автоматически самим внешним тестом - **ExtClient.exe**. После запуска внешнего теста открывается следующая форма:

Внешний тест-клиент

Подключение к объекту

Регистраторы:

- 402400076
- 402300005
- 402300002
- 402300004
- 402300009
- 402000012

Выбранный PPO: []

Подключиться к регистратору []

Errors

ErrorString []

FPErr []

X-отчет Z-отчет Отрезать чек Открыть ящик Отчет модема

Нефискальный чек

Текст комментария: Комментарий

Количество строк: 30

Выполнить

Периодический отчет по дате

Дата - начало: 11.09.2017

Дата - конец: 11.09.2017

Выполнить

Фискальный чек

Название артикула: Артикул 1

Налоговая группа: 0

Цена: 0.01

Количество: 1.000

Тип оплаты: Безнал 1

☐ Чек аннулировать

Код транзакции: 00000000000000000000000000000001

Выполнить

Вывод на дисплей

Очистка

Строка на дисплей

Выполнить

Пропуск строк: 1

Выполнить

Служебный внос/вынос

Сумма: 1.00

Выполнить

Тип

☒ Служебное внесение

☐ Служебное вынесение

Версия PPO Аннулировать или закрыть чек О программе Завершить

После запуска необходимо выбрать из списка **Регистраторы IDDEV** целевого регистратора и нажать кнопку **Подключиться к регистратору**. Как только выполнится подключение и квадрат находящийся справа от кнопки **Подключиться к регистратору** не окрасится в зеленый цвет. После этого можно выполнять действия на выбранном регистраторе.

Примечание. Выбирая ту или иную из предложенных операций необходимо соблюдать осторожность, чтобы выполнить каких-либо невозвратных действий на регистраторе (например, Z-отчет) в случае если он фискализирован.

7. Образцы для обработок в 1С 8

В комплект ПО входят примеры внешних обработок для 1С8.1 (**AtlasFPNet1.epf**) и 1С8.2 (**AtlasFPNet2.epf**). Ниже рассматриваются варианты использования методов COM объекта в процедурах этих обработок. Приведенный ниже код носит демонстрационный характер и может содержать неточности и нуждаться в уточнениях в зависимости от версии 1С, в которой он должен использоваться.

7.1. Функция **СоздатьОбъектДрайвера**

Основной код, используемый в функции –

```
ПрогиД="AtlasFPNet.FPregAtlas";           //ProgID COM объекта

Драйвер = Новый ComObject(ПрогиД);         //Создание объекта

ResultCode = Драйвер.OpenNew("FiscPrinter"); //Инициализация объекта

Результат1=Драйвер.NetConnect();           //Запуск TCP/IP сервера
```

В результате выполнения этих строк создается и инициализируется COM объект, а указатель на этот объект помещается в переменную **Драйвер**.

7.2. Функция **Подключить**

Основной код, используемый в функции –

```
Объект.Драйвер.Claim(20);

Объект.Драйвер.DeviceEnabled = True;

deRes = Объект.Драйвер.DeviceEnabled;

res = Объект.Драйвер.ResultCode;

If Объект.Драйвер.ResultCode = 114 Then Результат = мОшибкаНеизвестно Else res1
= "";

res2 = Объект.Драйвер.ErrorString;
```

```
Объект.Параметры.Свойство("IDDEV", времIDDEV);   ///извлекаемый параметр из  
//формы Настройки
```

```
EndIf;
```

```
Объект.Драйвер.ID_DEV=времIDDEV;   ///текущий ID_DEV
```

```
res3=Объект.Драйвер.GetRROConnection(времIDDEV);
```

```
if(res3 < 0)
```

```
    Message("Внимание! Номер регистратора отсутствует в списке разрешенных  
регистраторов!"); EndIf;
```

```
if(res3=0)
```

```
    Message("Внимание! Регистратор не присоединился к объекту!"); EndIf;
```

```
if(res3=10)
```

```
    Message("Внимание! Регистратор выбран текущим, но не присоединился к  
объекту!"); EndIf;
```

7.3. Процедура **Отключить**

Основной код, используемый в процедуре –

```
Объект.Драйвер.ResetPrinter();
```

```
Объект.Драйвер.NetDisconnect();
```

```
Объект.Драйвер.DeviceEnabled = False;
```

```
deRes = Объект.Драйвер.DeviceEnabled;
```

```
Объект.Драйвер.ReleaseDevice();
```

7.4. Функция **X-отчет**

Основной код, используемый в процедуре –

```
Объект.Параметры.Свойство("IDDEV", времIDDEV);
```

```
Объект.Драйвер.ID_DEV=времIDDEV;  
  
Перем res3;  
  
res3=Объект.Драйвер.GetRROConnection(времIDDEV);  
  
if(res3 < 0)  
    Message("Внимание! Номер регистратора отсутствует в списке разрешенных регистраторов!");  
    Возврат Результат;  
  
EndIf;  
  
if(res3=0)  
    Message("Внимание! Регистратор не приконектился к драйверу!");  
    Возврат Результат;  
  
EndIf;  
  
if(res3=10)  
    Message("Внимание! Регистратор выбран текущим, но не приконектился к  
драйверу!");  
    Возврат Результат;  
  
EndIf;  
  
Объект.Драйвер.PrintXReport();    //выполнить X-отчет
```

7.5. Функция Z-Отчет

Основной код, используемый в процедуре –

```
Объект.Параметры.Свойство("IDDEV", времIDDEV);  
Объект.Драйвер.ID_DEV=времIDDEV;  
res3=Объект.Драйвер.GetRROConnection(времIDDEV);
```

```
if(res3 < 0)
```

```
    Message("Внимание! Номер регистратора отсутствует в списке разрешенных  
    регистраторов!");
```

```
    Возврат Результат;
```

```
EndIf;
```

```
if(res=0)
```

```
    Message("Внимание! Регистратор не приконектился к драйверу!");
```

```
    Возврат Результат;
```

```
EndIf;
```

```
if(res=10)
```

```
    Message("Внимание! Регистратор выбран текущим, но не приконектился к драйверу!");
```

```
    Возврат Результат;
```

```
EndIf;
```

```
Объект.Драйвер.PrintZReport();
```

7.6. Функция **Сумма**

Предназначена для служебного внесения и вынесения денег.

Основной код, используемый в процедуре –

```
Объект.Параметры.Свойство("IDDEV", времIDDEV);
```

```
Объект.Драйвер.ID_DEV=времIDDEV;
```

```
Перем res3;
```

res3=Объект.**Драйвер.GetRROConnection**(времIDDEV);

if(res3 < 0)

Message("Внимание! Номер регистратора отсутствует в списке разрешенных регистраторов!");

Возврат Результат;

EndIf;

if(res3=0)

Message("Внимание! Регистратор не приконектился к драйверу!");

Возврат Результат;

EndIf;

if(res3=10)

Message("Внимание! Регистратор выбран текущим, но не приконектился к драйверу!");

Возврат Результат;

EndIf;

Объект.**Драйвер.ResetPrinter**();

if (Сумма >= 0) Объект.**Драйвер.FiscalReceiptType** = 1;

else Объект.**Драйвер.FiscalReceiptType** = 2;

EndIf;

Result = Объект.**Драйвер.BeginFiscalReceipt**(True);

if (Сумма < 0) Сумма=-Сумма; EndIf;

Result = Объект.**Драйвер.PrintRecCash**(Сумма);

Result = Объект.**Драйвер.EndFiscalReceipt**(True);

Возврат Результат;

7.7. Функция **ОткрытьЧек**

Объект.**Драйвер.ID_DEV**=времIDDEV;

Перем res3;

res3=Объект.**Драйвер.GetRROConnection**(времIDDEV);

if(res3 < 0)Message("Внимание! Номер регистратора отсутствует в списке разрешенных регистраторов!");

Возврат Результат;

EndIf;

if(res3=0)

Message("Внимание! Регистратор не присоединился к объекту!");

Возврат Результат;

EndIf;

if(res3=10)

Message("Внимание! Регистратор выбран текущим, но не присоединился к объекту!");

Возврат Результат;

EndIf;

If(ЭтоВозврат) Объект.**Драйвер.FiscalReceiptType** = 7;

Else

Объект.**Драйвер.FiscalReceiptType** = 4;

Endif;

if (ФискальныйЧек) Объект.**Драйвер.BeginFiscalReceipt**(True);

Иначе Объект.**Драйвер.BeginNonFiscal**();

КонецЕсли;

7.8. Функция **ПечататьСтроку**

Выполняется продажа (или возврат) товара.

Наименование1=Лев(Наименование,75);

if (НДС = 20) Then Налог=0;

elseif НДС = 0 Then Налог=1;

Else

 Объект.ОписаниеОшибки = НСтр("ru='Фискальный регистратор ';uk='Фіскальний реєстратор'") + НСтр("ru=' не правильная величина налога ';uk=' помилкове значення налогу'") + ОписаниеОшибки() + ".";

Результат = мОшибкаНеизвестно;

Возврат Результат;

EndIf;

 EndIf;

Результат = мНетОшибки;

if Объект.**Драйвер.PrinterState** = 9 Then //нефискальный чек

Объект.**Драйвер.PrintNormal**(2,Наименование); //печатать комментарий

//если же чек фискальный или возвратный -

Elseif Объект.**Драйвер.FiscalReceiptType** = 4 Или Объект.**Драйвер.FiscalReceiptType** = 7
Then

Объект.**Драйвер.PrintRecItem**(Наименование1,0,Количество*1000,Налог,Цена,"");

 Else

 Объект.ОписаниеОшибки = НСтр("ru='Фискальный регистратор ';uk='Фіскальний реєстратор'") + НСтр("ru=' ошибка состояния устройства ';uk=' помилка статусу реєстратора'") + ОписаниеОшибки() + ".";

Результат = мОшибкаНеизвестно;

Возврат Результат;

EndIf;

EndIf;

if Объект.Драйвер.ResultCode <> 0 Then

Объект.ОписаниеОшибки = НСтр("ru='Кассовый аппарат ';uk='Касовий апарат '" +
НСтр("ru=' ошибка печати/продажи ';uk=' помилка друку/продажу '" + ОписаниеОшибки()
+ ".");

Результат = мОшибкаНеизвестно;

Возврат Результат;

Endif;

if Объект.Драйвер.FiscalReceiptType = 4 Или Объект.Драйвер.FiscalReceiptType = 7 Then

if Скидка <> 0 Then

Объект.Драйвер.PrintRecltemAdjustment(?(Скидка > 0 , 3 , 4), "", ?(Скидка > 0 , Скидка , -
Скидка), 0);

EndIf;

EndIf;

Если Объект.Драйвер.ResultCode <> 0 Тогда

Объект.ОписаниеОшибки = НСтр("ru='Кассовый аппарат ';uk='Касовий апарат '" +
НСтр("ru=' ошибка скидки ';uk=' помилка знижки '" + ОписаниеОшибки() + ".");

Результат = мОшибкаНеизвестно;

Возврат Результат;

EndIf;

7.9. Функции **АннулироватьЧек** и **ОтменитьЧек**

Перем res3;

Объект.Параметры.Свойство("IDDEV", времIDDEV);

Объект.Драйвер.ID_DEV=времIDDEV;


```
Объект.Драйвер.ResetPrinter();  
  
    if Объект.Драйвер.ResultCode <> 0 Then  
  
        Объект.ОписаниеОшибки = НСтр("ru='Кассовый аппарат ';uk='Касовий апарат '" +  
НСтр("ru=' ошибка отмены чека ';uk=' помилка відміни чеку '" + ОписаниеОшибки() + ".");  
  
        Результат = мОшибкаНеизвестно;  
  
        Возврат Результат;  
  
Elseif;
```

7.10. Функция **ЗакрытьЧек**

```
If Объект.Драйвер.PrinterState = 9 Then  
  
    Объект.Драйвер.EndNonFiscal();  
  
If Объект.Драйвер.ResultCode <> 0 Then  
  
    Объект.ОписаниеОшибки = НСтр("ru='Кассовый аппарат ';uk='Касовий апарат '" +  
НСтр("ru=' ошибка оплаты ';uk=' помилка оплати '" + ОписаниеОшибки() + ".");  
  
    Результат = мОшибкаНеизвестно;  
  
    Возврат Результат;  
  
Elseif;  
  
Elseif;  
  
If СуммаБезнал <> 0 Then  
  
    Объект.Драйвер.PrintRecTotal(СуммаБезнал, СуммаБезнал, "1");  
  
    If Объект.Драйвер.ResultCode <> 0 Then  
  
        Объект.ОписаниеОшибки = НСтр("ru='Кассовый аппарат ';uk='Касовий апарат '" +  
НСтр("ru=' ошибка оплаты ';uk=' помилка оплати '" + ОписаниеОшибки() + ".");  
  
        Результат = мОшибкаНеизвестно;  
  
        Возврат Результат;
```

ElseIf;

ElseIf;

If СуммаНал <> 0 Или (СуммаНал = 0 И СуммаБезнал = 0)Then

Объект.**Драйвер.PrintRecTotal**(СуммаНал, СуммаНал, "0");

Если Объект.**Драйвер.ResultCode** <> 0 Then

Объект.ОписаниеОшибки = НСтр("ru='Кассовый аппарат ';uk='Касовий апарат '"') +
НСтр("ru=' ошибка оплаты ';uk=' помилка оплати '"') + ОписаниеОшибки() + ".";

Результат = мОшибкаНеизвестно;

Возврат Результат;

EndIf;

EndIf;

Объект.**Драйвер.EndFiscalReceipt**(True);

If Объект.**Драйвер.ResultCode** <> 0 Тогда

Объект.ОписаниеОшибки = НСтр("ru='Кассовый аппарат ';uk='Касовий апарат '"') +
НСтр("ru=' ошибка оплаты ';uk=' помилка оплати '"') + ОписаниеОшибки() + ".";

Результат = мОшибкаНеизвестно;

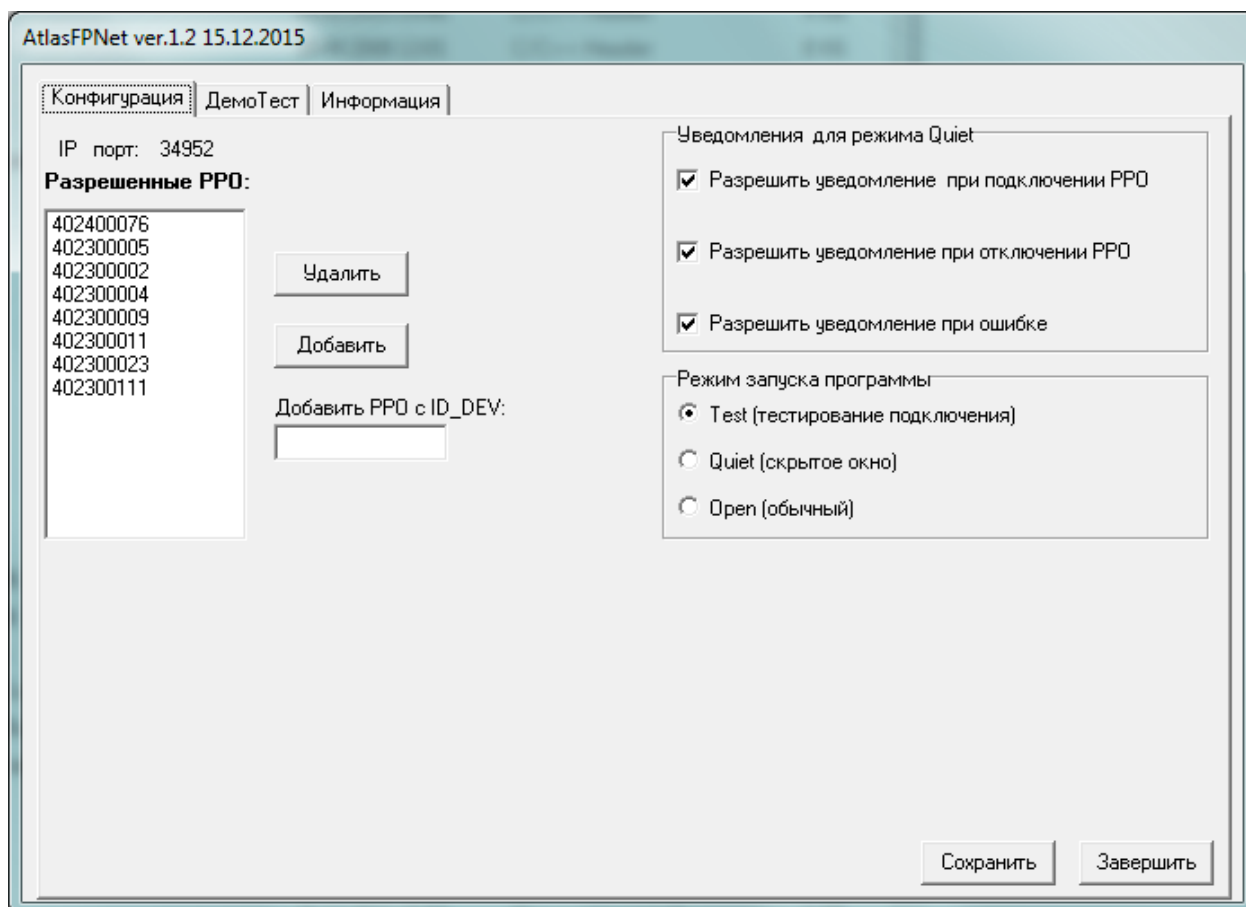
Возврат Результат;

EndIf;

8. Конфигурирование ПО

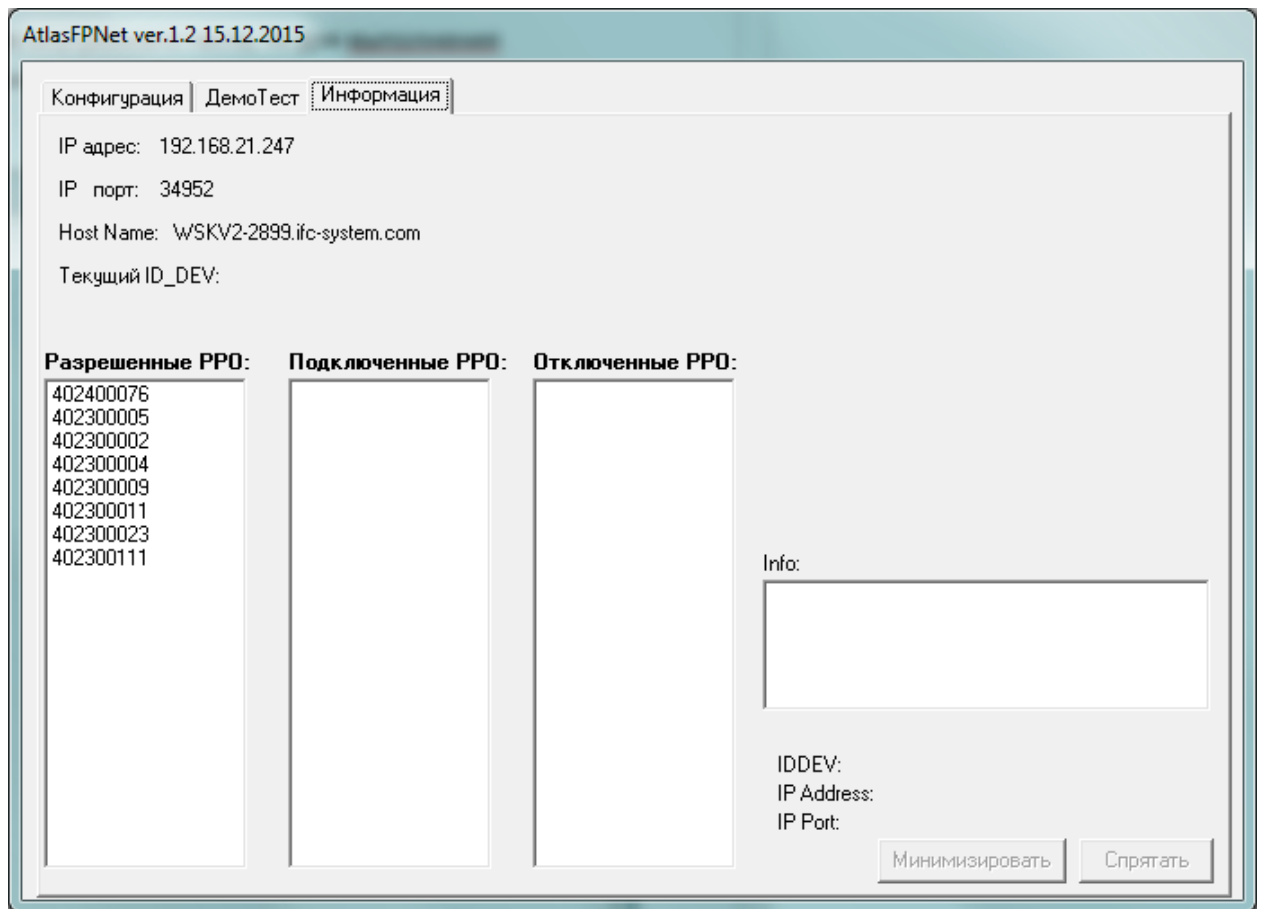
Конфигурирование ПО выполняется либо непосредственным редактированием файла **AtlasFPNet.ini**, либо запуском COM объекта в режиме внутреннего теста.

Если COM объект был запущен в режиме внутреннего теста, то для выполнения конфигурирования необходимо после запуска перейти на вкладку **Конфигурация**.



Три режима, поддерживаемые регистратором:

- 1) Внутреннего тестирования (**Test**). При указании этого режима и запуске COM объекта вручную, ПО сразу переходит в режим внутреннего тестирования.
- 2) Режим скрытое окно (**Quiet**). В этом режиме при вызове объекта, окно, соответствующее объекту, не появляется на экране. Однако если предварительно были настроены уведомления от сетевых событий в режиме Quiet, то информационное окно будет появляться при возникновении выбранных событий – подключение регистраторов, отключение регистраторов, сетевой ошибке.
- 3) Обычный режим (**Open**). При запуске объекта открывается форма объекта на вкладке **Информация**. Сетевые события, возникающие при работе объекта будут отображаться в этом окне.



В ini-файле выбору текущего режима соответствуют следующие строки:

[Start]

;Program=Quiet - для режима Quiet

[Start]

Program=Open - для режима Open

[Start]

Program=Test - для режима Test

В случае выбора режима **Quiet** значимыми становятся уведомления о сетевых событиях режима Quiet. В ini-файле выбору им соответствуют следующие строки:

[QuietNotification]

Connected=Yes

Disconnected=Yes

Error=Yes

Параметр **Connected**, установленный в **Yes** означает, что появление информации о подключении регистратора будет появляться в возникающем окне, а значение **No** означает, что окно-оповещение на событии подключения регистратора возникать не будет.

Параметр **Disconnected**, установленный в **Yes** означает, что появление информации о отключении регистратора будет появляться в возникающем окне, а значение **No** означает, что окно-оповещение на событии отключения регистратора возникать не будет.

Параметр **Error**, установленный в **Yes** означает, что появление информации об сетевой ошибке регистратора будет появляться в возникающем окне, а значение **No** означает, что окно-оповещение на событии сетевая ошибка регистратора возникать не будет.

В левой части конфигурационного окна перечень разрешенных РРО – список можно менять или в режиме работы с конфигурационным окном, или корректируя файл **ini**:

[FiscalPrinters]

IDDEV1=402400076

IDDEV2=402300005

IDDEV3=402300002

IDDEV4=402300004

IDDEV5=402300009

IDDEV6=402300011

IDDEV7=402300023

IDDEV8=402300111

9. Особенности запуска ПО

Главной особенностью и одновременно проблемой при работе с сетевыми устройствами является существенная асинхронность при взаимодействии с ними, связанная с особенностями TCP/IP протокола и взаимным местонахождением этих устройств.

Поскольку согласно выработанной концепции регистратор играет роль TCP/IP клиента, а СОМ объект – TCP/IP сервера, то именно СОМ объект находится в состоянии ожидания, а регистратор выступает в качестве активного устройства при подключении. Иными словами, СОМ объект перехватывает инициативу и начинает играть активную роль, посылая команды на регистратор, только после того как регистратор "постучался" к нему для открытия соединения.

Таким образом, структура работы с СОМ объектом следующая.

- 1) Внешнее ПО запускает СОМ объект
- 2) СОМ объект становится в состояние ожидания подключения к нему регистратора(ов).
- 3) До момента подключения нужного регистратора внешнее ПО может запрашивать статус подключения регистратора, кроме того оператор может визуально наблюдать за состоянием подключения (в обычном режиме СОМ объекта) или за всплывающими окнами в режиме (скрытого окна).
- 4) Как только подключение регистратора будет зафиксировано программным обеспечением, внешнее ПО может приступить к активному взаимодействию с регистратором посредством СОМ объекта.

Приложение. Дополнительные команды для работы с протоколом
Криптон вер. 5.03

STDMETHODIMP TFPregAtlasImpl::GetCommandParams(long regtype, long* Result)

В основе метода - команда 94 (0x5E)

входные данные:

int regtype – тип запроса. 0 – читать параметры первого запрограммированного товара;
1 – читать параметры следующего запрограммированного;
2 - читать параметры первого запрограммированного товара, по которому были продажи (возвраты);
3 – читать параметры следующего запрограммированного, по которому были продажи (возвраты).
Парами идут значения:
- сначала один раз 0, потом много раз 1;
- сначала один раз 2, потом много раз 3.
Если больше нет товаров, в ответе все данные в нуле.

выходные данные:

N	Описание	Тип	Размер	Примечание
1.	Код товара	N	6	
2.	Наименование товара	X	75	
3.	Налоговая группа	N	1	От 0 до 8
4.	Номер отдела	N	1	
5.	Модификатор	H	1	Таблица 3
6.	Единица измерения	S	5	Обозначение размерности
7.	Код УКТЗЕД	S	20	
8.	Цифровое значение штрих-кода	S	48	В версии 5.02 было тип – N, размер - 6
9.	Количество проданного товара	Q	8	
10.	Оборот по продажам	M	8	

11.	Количество возвратного товара	Q	8	
12.	Оборот по возвратам	M	8	

STDMETHODIMP TFPregAtlasImpl::MakeExciseSale(long code, BSTR quant, BSTR price, BSTR exc1, BSTR exc2, BSTR exc3, BSTR exc4, BSTR exc5, BSTR exc6, long* Result)

Продажа товара с акцизами.

В основе метода - команда 124 (0x7C)

входные данные:

int code - код товара

string quant - количество

string price - цена

string exc1 – акциз 1

string exc2 – акциз 2

string exc3 – акциз 3

string exc4 – акциз 4

string exc5 – акциз 5

string exc6 – акциз 6

выходные данные:

N	Описание	Тип	Размер	Примечание
1.	Сума продажи	M	4	

Описание функции:

По команде продается товар со специальной акцизной пометкой (для алкогольных товаров). Каждая единица такого товара должна иметь свой акциз. По данной команде можно продавать только целое количество товара и не более 6 единиц. Акцизы распечатываются перед данными по товару и передаются в налоговую.

**STDMETHODIMP TFPregAtlasImpl::MakeElektrPayment(long vid, BSTR sumop,
BSTR komis, BSTR idorgeq, BSTR idpu, BSTR eltypplat, BSTR platsys,
BSTR kodauto, BSTR kodtr, BSTR nazvplatop, BSTR refsign, long* Result)**

Безналичная оплата при непосредственном подключении банковского терминала к рабочему месту.

В основе метода - команда 123 (0x7B)

входные данные:

int vid - вид оплаты

string sumop - сумма оплаты

string komis - комиссия

string idorgeq - идентификатор организации эквайера

string idpu - идентификатор платежного устройства

string eltypplat - электронный тип платежа

string platsys - платежная система

string kodauto - код авторизации

string kodtr - код транзакции

string nazvplatop - Название платежной операции

string refsign - Указатель на подпись

SystemGroup Украина
ДП "Компания "ATLAS" ©
AtlasFPNet.exe. Руководство по эксплуатации.

Приклад чеку		
ТОВ «СИСТЕМ ПРОДАКШН»		
http://www.systemgroup.com.ua		
ПН 123456789012		
Касир N001 Каса 01		

0.720 X 51.45=		
Ковбаса Лікар.ГРОШИНО	37.04	A
0.720 X 51.45=		
Ковбаса Лікар.ГРОШИНО	37.04	A
Акульовано		
0.720 X 51.45=		
Ковбаса Лікар.ГРОШИНО	-37.04	A
Батон "МІДІА"В/Г	2.80	I
20.000 X 4.60=		
Листівка вітальна	93.60	B
Надбавка 10.00%	9.36	B
5.000 X 21.39		
Гель д/сантех.САЙМА	106.95	ED
Надбавка на суму	5.00	*
Звонки -10.00% на податки. групу	-3.78	BA
Пиво Leff 0.3	38.60	OK
AB BP 26 AADB 778549		
AB BP 26 AADB 778550		
2.000 X 70.00		
Код УКТЗЕД 22071000		
Штрих код 2000000022070		
Кованки 0.5	140.00	GB

Сума без ПДВ	429.57	
ПДВ Ж = 20.00%	7.72	

АКЦ.ЗВ. И = 5.00%		2.32

ОщадБанк		
Термінал 80000001		
Комісія		2.00
Сплата		
ЕПС **** * 1234		
ПЛАТІЖНА СИСТЕМА MasterCard		
КОД ТРАНЗ. 123456789012		
КОД АВТ. 123456		
КАСІР -----		
ДЕРЖАТЕЛЬ ЕПС -----		
Картка	100.00	грн
Готівка	350.00	грн
Решта	-10.40	

Сума	439.61	
ПДВ А = 20.00%	5.67	
ПДВ В = 20.00%	22.22	
АКЦ. ЗВ. Г = 5.00%	6.67	
ПДВ Д = 20.00%	17.88	
ПДВ Є = 2.00%	1.79	
ПДВ Ж = 20.00%	7.72	
АКЦ. ЗВ. И = 5.00%	2.32	

ЗАКРУГЛЕННЯ	-0.01	грн
до сплати	439.60	

000003172 00001	20.11.2020	9:31:02
		
ЗН СП802000001		
ФН 1234567890		
ФІСКАЛЬНИЙ ЧЕК		Atlas

выходные данные:

N	Описание	Тип	Размер	Примечание
1.	Остаток для оплаты	M	4	

Описание функции:

Предназначена для передачи в PPO, всех параметров оплаты внешнего электронного платежа.

STDMETHODIMP TFPregAtlasImpl::PayCash2(long code, BSTR sumvip, BSTR komis, BSTR idorgeq, BSTR idpu, BSTR eltypplat, BSTR platsys, BSTR kodauto, BSTR kodtr, long* Result)

Только для версии протокола Криптон 5.02

В основе метода - команда 122 (0x7A)

входные данные:

int numop - вид оплаты

string sumvip - сумма оплаты

string komis - комиссия

string idorgeq - идентификатор организации эквайера

string idpu - идентификатор платежного устройства

string eltypplat - электронный тип платежа

string platsys - платежная система

string kodauto - код авторизации

string kodtr - код транзакции

выходные данные:

N	Опис	Тип	Розмір	Примітка
1.	Результат	M	4	

Описание функции:

- Данная команда не требует перед ее выполнением открытия чека и закрытия чека после ее выполнения (эти операции автоматически выполняются самой командой).

STDMETHODIMP TFPregAtlasImpl::OpenPayCash3(long* Result)

Только для версии протокола Криптон 5.03

Используется для открытия чека выдачи денег (PayCash3)

**STDMETHODIMP TFPregAtlasImpl::PayCash3(BSTR sumvip, BSTR komis,
BSTR idorgeq, BSTR idpu, BSTR eltypplat, BSTR platsys, BSTR kodauto,
BSTR kodtr, long* Result)**

Только для версии протокола Криптон 5.03

В основе метода - команда 122 (0x7A)

входные данные:

string **sumvip** - сумма оплаты

string **komis** - комиссия

string **idorgeq** - идентификатор организации эквайера

string **idpu** - идентификатор платежного устройства

string **eltypplat** - электронный тип платежа

string **platsys** - платежная система

string **kodauto** - код авторизации

string **kodtr** - код транзакции

выходные данные:

N	Опис	Тип	Розмір	Примітка
1.	Результат	M	4	

Описание функции:

- Данная команда требует, чтобы перед ее выполнением была выполнена команда Открытие фискального чека (63h) с параметром 16 (чек выплаты наличных держателям ЕПС). После выполнения операции необходимо выполнить команду 65h, которая закрывает чек.

STDMETHODIMP TFPregAtlasImpl::CloseCash3(long* Result)

Только для версии протокола Криптон 5.03

Используется для закрытия чека выдачи денег (PayCash3)

STDMETHODIMP TFPregAtlasImpl::TestTCPconn(long* Result)

Тестирование сетей (эквайер и т.д.)

В основе метода - команда 151 (0x97)

входные данные:

char *psw – пароль. В качестве пароля можно использовать шестизначное произвольное число (рекомендуется - 000000).

выходные данные:

int* error – код ошибки (не используется)

N	Описание	Тип	Размер	Примечание
1	Адрес и порт сервера эквайера	S		IP или DNS адрес:порт
2	Результат тестирования соединения эквайера	N	1	0-соединение не было установлено 1-соединение было установлено
3	Адрес и порт сервера удаленного доступа	S		IP или DNS адрес:порт
4	Результат тестирование соединения сервера удаленного доступа	N	1	0-соединение не было установлено 1-соединение было установлено
5	Адрес и порт сервера cashandcloud	S		IP или DNS адрес:порт

6	Результат тестирования соединения сервера cashandcloud	N	1	0-соединение не было установлено 1-соединение было установлено
---	--	---	---	---

Описание функции:

Пример ответного сообщения

192.168.22.165:50000;0;rro.dp-atlas.com.ua:34954;1;rro.cashandcloud.com.ua:55999;1;

Примечание:

IP или DNS адрес запроса выбирается в соответствии с текущими настройками фискального регистратора

```
STDMETHODIMP TFPregAtlasImpl::PrintRecItemNew(BSTR description,  
        CURRENCY price, long quantity, long vatInfo, CURRENCY unitPrice,  
        BSTR unitName, BSTR uktzed, BSTR barcode, long* Result)
```

Выполняет те же действия, что и **PrintRecItemNewBSTR**, но в качестве цены принимает плавающее число, а количество указывается в граммах (в виде целого числа) .

```
STDMETHODIMP TFPregAtlasImpl::ReadArticulCod(long code, long* Result)
```

В основе метода - команда 89 (0x59)

Чтение информации о товаре с данным кодом.

входные данные:

char *psw – пароль. В качестве пароля можно использовать шестизначное произвольное число (рекомендуется - 000000).

int code - код товара

выходные данные:

N	Описание	Тип	Размер	Примечание
1.	Код товара	N	6	
2.	Наименование товара	X	75	
3.	Налоговая группа	N	1	От 0 до 8
4.	Номер отдела	N	1	
5.	Модификатор	N	1	Таблица 3
6.	Единица измерения	S	5	Обозначение размерности
7.	Код УКТЗЕД	S	20	
8.	Цифровое значение штрих-кода	S	48	В версии 5.02 было тип – N, размер - 6

Описание функции:

Предначена для чтения запрограммированного товара из регистратор (в версии для протокола 5.03 добавлены поля код УКТЗЕД и цифровое значение штрих-кода)

STDMETHODIMP TFPregAtlasImpl::PrintQRcode(BSTR code, long sizeX, long* Result)

Печать QR-кода

В основе метода - команда 228 (0xE4)

входные данные:

char *psw – пароль. В качестве пароля можно использовать шестизначное произвольное число (рекомендуется - 000000).

char* code - текстовая строка для штрих кода (105 символов)

int size - размер купона (От 0 до 10 (По умолчанию: 4), 1 – наименьший, 10 – наибольший, 0(4) – умолчание)

выходные данные:

Описание функции:

Печать двумерного штрих кода (QR - код)

STDMETHODIMP TFPregAtlasImpl::QRInfo(long oper, BSTR Data, long* Result)

В основе метода - команда 229 (0xE5)

входные данные:

int oper - команда

char* data - информация

выходные данные:

SystemGroup Украина
ДП "Компания "ATLAS" ©
AtlasFPNet.exe. Руководство по эксплуатации.

N	Описание	Тип	Размер (байт)	Примечание
1.	Информация	S	50	

Описание функции:

Данная команда используется для вывода некоторой запрограммированной информации в рамках чека (например, ссылку на сайт) с необходимым по закону QR-кодом. Информация сохраняется в батарейной памяти, поэтому может как сохраняться после выключения, так и часто перезаписываться, например, под каждый чек.

STDMETHODIMP TFPregAtlasImpl::ReadQRExt(long* Result)

В основе метода - команда 230 (0xE6)

входные данные:

char *psw – пароль. В качестве пароля можно использовать шестизначное произвольное число (рекомендуется - 000000).

выходные данные:

int* error – код ошибки (не используется)

Описание функции:

Возвращает данные обязательного QR-кода, формируемого по итогам чека.

Пример: FN0000000023 N9 =2.00 10.07.2020 11:54
MAC=ABoAAAHBAAAAAAMAAABAACvkgQCF/YljJqgkAY=

**STDMETHODIMP TFPregAtlasImpl::PrintRecExciseItemBSTR(BSTR description,
BSTR price, BSTR quantity, long vatInfo, BSTR unitPrice, BSTR unitName,
BSTR uktzed, BSTR barcode, BSTR ex1, BSTR ex2, BSTR ex3, BSTR ex4,
BSTR ex5, BSTR ex6, long* Result)**

Программирование и продажа акцизного товара.

входные данные:

`string description` – название товара

`string price` – не используется

`string quantity` – количество , разделитель точка, три знака послеточки обязательные.

`long vatInfo` – номер налога (0-9), налоги должны быть предварительно запрограммированы

`string unitPrice` – цена товара за единицу, разделитель точка, два знака послеточки обязательные.

`string unitName` – до пяти символов название единиц товара: "шт." "кг"

`string uktzed` – код УКТЗЕД

`string barcode` – штрихкод

`string exc1` – акциз 1 (номер-штрихкод 1 марки)

`string exc2` – акциз 2 (номер-штрихкод 2 марки)

`string exc3` – акциз 3 (номер-штрихкод 3 марки)

`string exc4` – акциз 4 (номер-штрихкод 4 марки)

`string exc5` – акциз 5 (номер-штрихкод 5 марки)

`string exc6` – акциз 6 (номер-штрихкод 6 марки)

Название товара хешируется в номер, который используется при программировании товара в регистратор, после этого акцизный товар продается. При этом (!!!) команда позволяет продать от 1 до 6 единиц товара сопровождая их номерами марок. Таким образом количество может быть: "1.000", "2.000", "3.000", "4.000", "5.000", "6.000". Если продается количество менее 6 единиц, то марки заполняются последовательно без пропусков, оставшиеся места заполняются пустыми строками.

**STDMETHODIMP TFPregAtlasImpl::PrintRecItemNewBSTR(BSTR description,
BSTR price, BSTR quantity, long vatInfo, BSTR unitPrice, BSTR unitName,
BSTR uktzed, BSTR barcode, long* Result)**

Программирование и продажа товара.

Продажа товара с акцизами.

входные данные:

`string description` – название товара

`string price` – не используется

`string quantity` – количество, разделитель точка, три знака послеточки обязательные.

`long vatInfo` – номер налога (0-9), налоги должны быть предварительно запрограммированы

`string unitPrice` – цена товара за единицу, разделитель точка, два знака послеточки обязательные.

`string unitName` – до пяти символов название единиц товара: "шт." "кг"

`string uktzed` – код УКТЗЕД

`string barcode` – штрихкод

Название товара хешируется в номер, который используется при программировании товара в регистратор, после этого акцизный товар продается.

**STDMETHODIMP TFPregAtlasImpl::PrintRecExciseItem(BSTR description,
CURRENCY price, long quantity, long vatInfo, CURRENCY unitPrice,
BSTR unitName, BSTR uktzed, BSTR barcode, BSTR ex1, BSTR ex2, BSTR ex3,
BSTR ex4, BSTR ex5, BSTR ex6, long* Result)**

Выполняет те же действия, что и **PrintRecExciseItemBSTR**, но в качестве цены принимает плавающее число, а количество указывается в граммах как целое значение.