

Модуль работы с турникетами, соляриями

Оборудование, настройка модуля взаимодействия с турникетами, управление соляриями.
Описание работы с устройствами PocketKey (биометрия, турникеты, солярий)

- [Турникеты](#)
- [Контроль доступа с помощью биометрии. Интеграция с PocketKey](#)
- [Модуль "Управления турникетом"](#)
- [Подключение CardLan как считывателя для турникета](#)
- [Мобильный турникет](#)
 - [Описание функционала](#)
 - [Техническая часть и настройки](#)
- [Проводные контролеры - изменение параметров и прошивка](#)
- [Описание контроллера](#)

Турникеты

Перечень устройств, с которыми работаем:

Турникеты производства Perco.

Триподы: Серии TTR-04, T-5 и т.п.



Тумбовые турникеты. Серия TTD



Скоростные проходы. ST-01



Как правило схема управления турникетом у них одинаковая.

Турникеты Praktika Т - 01



Турникеты Ростов-Дон т9м1.

Подключение идет через плату XS2



Ростов-Дон T283M1



Контроль доступа с помощью биометрии.

Интеграция с PocketKey

Для осуществления контроля биометрии используется интеграция с оборудованием и ПО PocketKey !!!

Данный функционал доступен с версии gkhostconnect 1.86.0.5 и новее.

1. Типовая логика обслуживания клиента.

- Клиента регистрируют в системе Абонемент
- Клиенту в системе Абонемент прикрепляется идентификатор (бесконтактная карта Mifare)
- Происходит обучение системы распознавания лицу данного клиента (см п.4)
- Клиент проходит к турникету на вход и прикладывает карту к считывателю
- Трек с карты передается в gkhostconnect для анализа доступа
- В случае разрешенного доступа gkhostconnect Отправляет в систему распознавания id клиента и происходит сверка фото в БД системы распознавания и фото с камеры на турникете
- В случае совпадения лица турникет открывается на вход
- При выходе контроль биометрии не осуществляется

2. Установка и настройка

Установку и настройку ПО для распознавания и сохранения биометрических данных устанавливается на компьютер заказчика специалистами PocketKey

В **gkhostconnect.ini** необходимо в секции **[gkhost]** прописать следующие настройки:

```
[gkhost]
host=172. 22. 3. 241
port=8075
local=0
CanSendData=1
```

```
ProtocolType = 3
```

Где:

- host, port - адрес и порт, на котором работает ПО PocketKey
- ProtocolType=3 - указывает, что необходимо работать с биометрией

Вот это нужно добавить в мануал:

-добавить в мануал указание настройки использовать биометрию

-айди устройств должны быть двузначными

-обучение проходит через веб fit.recognitionx.com, для того чтобы войти - в файле hosts прописать: 192.168.0.100(айпишник сервера где развернут покет кей локально)
fit.recognitionx.com

Всвязи с ограничениями связанными с хранением биометрических данных в PocketKey появилась реализация PocketKey Dynamic, в которой вместо биометрии используется моб. приложение (есть реализации на iOS и Android) и терминалы сканируют QR код из приложения (постоянно обновляется).

Внесение идентификатора в базу PocketKey просходит либо через личный кабинет РК либо автоматически при помощи предлагаемого со стороны PocketKey API (В Абонемент данное Api не реализовано).

Модуль "Управления турникетом"

Интерфейс "Абонемент-Турникет" - программный комплекс, позволяющий связать аппаратную часть (турникеты, магнитные замки и другие устройства управляемые с помощью контроллеров UCS) и ПО Абонемент, позволяя контролировать доступ клиентов к тем или иным платным зонам объекта.

Информация в данной статье актуальна для версий: *gkHost.exe* - 3.2.0.1 и *gkhostconnect.exe* - 1.88.0.6 , доступных в версии Абонемент 1.128 в папках `\FullVersionVer128\GKHOST\` и `\FullVersionVer128\gkhostconnect\` соответственно.

Введение

Комплекс состоит из 2х составляющих:

gkHost - отвечает за сообщение с контроллерами UCS;

gkhostconnect - отвечает за связь с БД Абонемент и исполнение установленной логики контроля доступа к указанным ресурсам.

Данные составляющие могут быть разнесены на разные машины. Связь между ними осуществляется в рамках протокола TCP-IP.

Но, как правило, удобно их держать в рамках основного сервера Абонемент.

gkHost

Данная составляющая должна быть установлена на машине, обязательно находящейся в той же подсети что и контроллеры UCS (по умолчанию - 172.31.x.x)(например, в рамках альтернативной конфигурации сетевого адаптера).

Принципиально важно, что бы подсеть была с маской 255.255.0.0. Адрес интерфейса, на котором gkHost работает, должен быть, в их случае, от 192.168.0.1 до 192.168.31.254.

Программа не имеет графического интерфейса.

Настройки:

gkHost.ini

```
[ log]
;Настройки логирования
Name = gk3Host.log
level = 10
debugLevel = 10
ByMonthes=1
ByDays=1
ByHours=1

[ log.protocol]
;Настройки логирования сообщения с контроллерами
Name = gk3Host.protocol.log
level = 10
debugLevel = 10
stampKind = 4145
ByMonthes=1
ByDays=1
ByHours=1

[ telnet]
;Порт используемый для связи с gkhostconnect
port = 5556

[ gk3]
;Представлены настройки подходящие, для контроллеров настроенных по умолчанию
;IP-адрес, на котором будет работать сервер протокола GK3 - должен совпадать с IP-адресом
машины в подсети контроллеров UCS
ip = 172.31.32.100 // здесь должно быть 172.31.0.100
```

```
;PORT -- порт протокола GK3
port = 18251
;NET -- сеть протокола GK3
net = 172.31.0.0
;MASK -- маска сети протокола GK3
mask = 255.255.0.0
```

Для установки процесса службой Windows, достаточно просто запустить прилагаемый *install_gkh.bat* от имени администратора.

gkhostconnect

Данная составляющая отвечает за связь с БД Абонемент и исполнение установленной логики контроля доступа к указанным ресурсам, как было указано выше.

Основные настройки в файле *gkhostconnect.ini*:

```
[AppService]
;Настройки службы создаваемой при запуске с ключом -install
;Имя сервиса
DisplayName = UCS Служба GKHOSTConnect
;Внутреннее имя сервиса (одно слово. Только латинские символы и цифры. Начинается с буквы)
ServiceName = UCSGKHOSTConnect
Description = Служба-интерфейс контроля доступа UCSGKHOSTConnect

[Application]
;Скрывать приложение автоматически, если запущено не службой
AutoHide=0

[log]
;Настройки логирования
name=. \log\gkhostconnect.log
level=30
debuglevel=30
ByMonthes=1
ByDays=1
```

```
BYHOURS=1
```

```
[ LOCK. SERVER]
```

```
;настройки на сервер блокировок Абонемент
```

```
TCPHOST = 127. 0. 0. 1
```

```
TCPPort = 1111
```

```
[ database]
```

```
;Настройки на БД Абонемент
```

```
dbname=127. 0. 0. 1/3080: C: \_UCS\_FITNESS\IBDATA\fitness. FDB
```

```
dbuser = ucs
```

```
DBPassword = oQKH8n0BFUE=
```

```
[ devices]
```

```
;Настройки применяемые для всех устройств
```

```
;Тарификацию начинаем после того как турникет провернулся:
```

```
при CheckTurnDirect=0 - считается, что турникет провернулся в ту сторону, в которую была дана ко
```

```
при CheckTurnDirect=1 - (по умолчанию) дополнительный контроль. Тарификация начинается , только
```

```
CheckTurnDirect=1
```

```
;Время ожидания прохода в секундах (должно быть больше, чем время ожидания в настройках турникет
```

```
TurniquetTimeOut=10
```

```
;Обрабатывать все символы в приходящих с контроллеров треках
```

```
ToProcessAllCharsInCard=1
```

```
; Не обрабатывать следующие треки ( CF = ' No Card' )
```

```
NoProcessedCardsNo=CF
```

```
[ gkhost]
```

```
;настройка связи с gkHost
```

```
host=127. 0. 0. 1
```

```
port=5556
```

```
local=0
```

```
ProtocolType=0
```

```
CanSendData=1
```

Остальные секции данного *ini* файла специфические и требуют контроля только при использовании специфического оборудования, и описаны в статьях посвящённых данному оборудованию.

При настройке базового функционала, изменять данные секции не требуется.

Для запуска программы в режиме графического интерфейса, запускать её требуется с ключом **-desktop**.

Для установки программы в качестве службы Windows, требуется запустить её с ключом **-install** от имени администратора.

Основное окно программы:

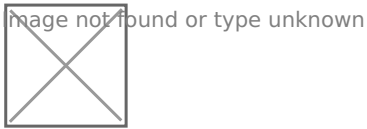


Рис.1 Основное окно программы *gkhostconnect*

Как видно на рис.1, основное окно программы содержит 3 основных вкладки:

GKHOST - настройки взаимодействия с *gkHost*;

Устройства контроля доступа - настройки взаимодействия с контроллерами UCS;

Печатные формы - печатные формы, обрабатывающие при различных событиях.

Также внизу окна отображается состояние связи с БД Абонемент и связи с *gkHost* (Пиктограмма лампочки в строке состояния: лампочка горит - связь установлена, лампочка погашена - связь не установлена).

Вкладка GKHOST:



Рис.2 Вкладка GKHOST основного окна *gkhostconnect*

На рис.2 представлена вкладка GKHOST. На данной вкладке, по умолчанию, настроек производить не требуется, если был корректно настроен *gkhostconnect.ini*

Вкладка Устройства контроля доступа:

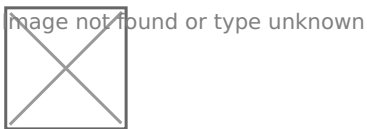


Рис.3 Вкладка Устройства контроля доступа

Вкладка Устройства контроля доступа представлена на рис.3. На данной вкладке происходит основной этап настройки *gkhostconnect* на конкретные контроллеры и логики контроля доступа к конкретным ресурсам в Абонемент.

Как видно на рис.3:

В левой части окна находится таблица со всеми ресурсами заведёнными в Абонемент;

В центральной части окна находится таблица со всеми контроллерами, которыми будет управлять *gkhostconnect*;

В правой части окна находятся свойства выбранного контроллера.

Соответственно, для внесения нового контроллера в список, требуется:

- Выбрать ресурс к которому будет привязан данный контроллер в левой части окна.

- Нажать кнопку "+" в центральной части окна.
- Далее произвести настройку свойств добавленного контроллера в правой части окна.
- Сохранить внесённые изменения кнопкой "✓" или кнопкой **"Сохранить"**.

Основные свойства контроллера:

- **device, relay** - предпоследняя и последняя цифра IP-адреса контроллера в десятичном формате. (например device=32 и relay=20, для IP=172.31.32.20);
- **Тип устройства** - указывает, какое устройство подключено к контроллеру;
- **Направление прохода** - направление прохода контролируемое данным контроллером;
- **Кого пускать** - самоговорящее название свойства;
- **Проход группы** - разрешён ли групповой проход для одного клиента, если есть таким образом настроенные абонементы.

Из остальных свойств отметим также следующие:

- **Генерировать событие при разрешённом доступе** - устанавливается в случае, если устройство подключённое к контроллеру не возвращает факт прохода. В таком случае, проход будет считаться осуществлённым в момент разрешения прохода и человек будет введён в ресурс, независимо от того, прошёл ли он;
- **Команда для входа, Команда для выхода, Команда для входа/выхода** - команды посылаемые на контроллер, для выполнения соответствующих действий. Определяются прошивкой контроллера. (Значения по умолчанию - для импульсного разрешения прохода для прошивки *gk3_trn_3_0_0_19_9600x2_hex*).

Все остальные свойства имеют говорящие названия, и настраиваются согласно схеме обслуживания клиентов.

Важно! Декодирование треков карт

Для декодирования треков приходящих со считывателей, подключённых к контроллерам, *gkhostconnect* использует *decodecards.dll*, настраиваемый через *decodecards.ini*.

Так как треки приходящие с контроллеров часто не совпадают в точности с треками получаемыми со считывателей на рабочих местах, требуется дополнительно поризвести тестирование и настройку декодирования.

Соответственно, использовать *decodecards.ini* настроенный под Абонемент или TCPCardReader не получится - декодированные треки будут отличаться от заведённых в Абонемент, и gkhostconnect не сможет найти клиента в БД.

Подключение CardLan как считывателя для турникета

- Контроллер gk является ведомым при таком подключении.

1) прошить контроллер спец. прошивкой (например 100)

2) в EEPROM контроллера прописать адрес CardLan'a

- CardLan подключить по microUSB (центральный вход сзади) и залить в него ethernet.sh с IP, MASK'ой и MAC'ом

IP=172.31.32.12

MASK=255.255.0.0

MAC=00:0E:C6:87:72:12

из инструкции от GK

> По прошивке контроллера для Cardlan.

>

> Нужно прошить прошивку в контроллер и установить ip адрес валидатора, с которым будет работать контроллер.

>

> Как это делается:

> В gkTools подключиться к контроллеру.

> Перейти во вкладку EEPROM и подгрузить данные(зеленый кружок).

> Перевести адрес валидатора, с которым будет работать устройство, в hex вид, например: 172.31.32.10(DEC) -> AC.1F.20.0A(HEX).

> Записать последовательно 4 значения адреса в поля с координатами 13,0 14,0 15,0 16,0 (это первая строка таблицы).

> Сохранить изменения(дискета).

> Перезагрузить контроллер.

>

> Внимание! Теперь контроллер возможно будет перезагрузить только снятием питания, тк

UDP порт изменился.

Мобильный турникет

Описание функционала

(реализованный и запланированный функционал)

Какие задачи решает приложение?

Приложение предназначено для контроля и учёта посещений гостей, для регистрации услуг гостю, для анализа загрузки ресурсов.

Как работает?

Приложение устанавливается на смартфон с NFC считывателем.

Приложение используют сотрудники предприятия (операторы), которые контролируют вход/выход в одну или несколько зон тарификации.

Для этого необходимо просканировать электронный идентификатор клиента и отметить вход/выход посетителя.

Для аналитики используется отчёт «Загрузка ресурсов».

Целевая аудитория

Любое предприятие, где требуется вести контроль и учёт посещений клиентов: например, спортивные площадки, детские развивающие и обучающие центры и пр.

Терминология

Пользователь – сотрудник объекта и пользователь системы Абонемент, который работает с мобильным приложением.

Посетитель – клиент, который посещает ресурсы и получает услуги.

Идентификатор – электронный носитель, прикрепленный к профилю посетителя или пользователя. При помощи идентификатора осуществляется авторизация пользователя или идентификация посетителя в приложении.

Профиль – учётная запись пользователя или посетителя.

Магазин – через интерфейс магазина пользователь может зарегистрировать (добавить без оплаты) посетителю дополнительные услуги.

Дополнительные услуги – заведенные услуги в справочнике услуг ПО Абонемент. К дополнительным услугам не относятся абонементы.

Начисление дополнительной услуги – регистрация дополнительной услуги в профиле посетителя без принятия денежных средств с последующей оплатой, например, на кассе.

Функциональные возможности приложения

Авторизация пользователя в приложении при помощи ввода логина/пароля или идентификатора;

Поиск посетителя по идентификатору;

Отображение информации о посетителе: ФИ, Фото, Ресурсы (доступные, а также те, в которых присутствует посетитель);

Регистрация входа посетителя (в один или несколько ресурсов);

Регистрация выхода посетителя (из одного или нескольких ресурсов);

Отображение тарифов по абонементу (если посещение в ресурс доступно по нескольким тарифам) в профиле посетителя;

Выбор тарифа, в рамках которого происходит регистрация посещения;

Отображение даты окончания действия абонемента в профиле посетителя;

Отображение списка дополнительных услуг в Магазине;

Возможность выбрать дополнительные услуги и начислить в профиле посетителя;

Возможность удалять дополнительные услуги из корзины (до момента начисления услуг);

Возможность регистрации дополнительных услуг в профиле посетителя;

Подтверждение идентификатором посетителя при начислении дополнительных услуг в профиле посетителя;

Навигация внутри магазина (НАЗАД, НАЧАЛО, В КОРЗИНУ);

Вывод информации о выбранных дополнительных услугах в магазине (КОЛ-ВО, СУММА);

Отчет «Загрузка ресурсов»:

Отображение количества присутствующих посетителей в ресурсах плюс остаток времени посещения;

Цветовая визуализация (красный цвет) в случае, если в ресурсе есть посетитель с превышением по времени посещения;

Цветовая визуализация (зеленый цвет) в случае, если в ресурсе есть посетитель;

Детализация по посетителям в рамках выбранного ресурса;

Обновление данных в автоматическом или ручном (кнопка ОБНОВИТЬ ОТЧЁТ) режимах;

Настройки приложения:

Настройка списка контролируемых пользователем ресурсов (дополнительно проверяются права пользователя для работы с ресурсами);

Отображение информации о пользователе (ФИ сотрудника, время авторизации, выход);

Отображение технической информации (адрес сервера, API ключ (зашифрованная информация)).

Необходимые пользовательские настройки

Справочник пользователей

Для успешной авторизации пользователя в приложении:

Пользователь должен быть заведен в справочнике пользователей.

Пользователю должен быть назначен логин и пароль для WEB (настройки/пользователи и роли/пользователи/карточка клиента/ поля «Логин для WEB», «Пароль для WEB»).

Для успешной авторизации пользователя в приложении при помощи идентификатора к профилю пользователя должен быть прикреплен идентификатор поддерживаемого типа.

Проверяется права доступа у пользователя для работы с ресурсами, а именно настройка «Доступные корпуса и этажи в контроле присутствия». Если ресурс недоступен согласно настройкам прав доступа, то он не отображается в списке ресурсов в профиле пользователя в приложении (раздел «Настройки/Ресурсы»). Это значит, что пользователь не может обслуживать посетителей в недоступных ему ресурсах.

Справочник услуг

Для отображения услуг в разделе Магазин необходимо настроить справочник услуг. В приложении отображаются все услуги, заведенные в справочнике.

Посетители

Для успешного поиска клиента в приложении:

Гость должен быть заведен в базе данных Абонемент.

Гостю должен быть прикреплен идентификатор поддерживаемого типа.

Для доступа в ресурс у гостя должны быть соблюдены условия посещения (например, период действия абонемента, расписание, тариф, оплата и пр.).

Техническая часть и настройки

Системные требования

Версия Абонемент:

- Не ниже 138

Мобильный телефон:

- Android 7.1+
- Для работы считывателя бесконтактных карт необходимо наличие NFC модуль

Требования к бесконтактным картам:

- Считывание производится только бесконтактных карт/браслетов формата Mifare Classic (4 byte UID)

Лицензирование

Модуль лицензируется по количеству одновременно работающих приложений.

Приложение занимает лицензию, пока в нём авторизован пользователь. Как только пользователь выходит из учётной записи в приложении занимаемая лицензия освобождается. Также лицензия освобождается автоматически, в случае бездействия пользователя в приложении в течении 15 минут.

[Подробнее про лицензирование модулей Абонемент.](#)

Настройки

Connector

Для работы приложения необходимо развернуть API (Connector) на любом доступном с мобильных устройств шлюзе.

Помимо обычной настройки Connector на БД, сервер блокировок, внешний порт доступный из сети, в которой будут находиться мобильные устройства, необходимо соблюсти следующие моменты:

В папке с connector должны присутствовать следующие файлы, доступные в поставке Абонемент версии 139 и выше:

- \Connector\ResponseTransformSchema.ini
- \Connector\XSLT\..
- \Connector\frf\MobTurn_ResourceLoading.fr3
- \Connector\frf\MobTurn_ResourceLoading.ini

В *connector.ini* следующие параметры **обязательно** должны быть заполнены таким образом:

```
[ charset ]
UseUrlEncode=0

[ DataFormat ]
RequestDataFormat=1
```

Для корректного считывания карт Mifare Classic в *decodecards.ini* используемого connector, настройки должны быть следующие:

```
[ general ]
usemask=1

[ mask ]
card_ALL = *

[ cards.card_ALL ]
code=2
CardDecodeType = MASK
mask=hhhhhhh
bitmask=$FFFFFFFF
```

Мобильное приложение

Для установки мобильного приложения достаточно передать файл .apk (установочный файл android) на устройство, и используя любые средства установить приложение.

В мобильном приложении необходимо на вкладке настроек "Технические" (пиктограмма шестерёнок) настроить адрес к настроенному API connector, и зашифрованный пароль доступа.

Шифрование пароля (PrivateKey connector.ini)

Для получения пароля доступа указываемого в мобильном приложении, необходимо воспользоваться утилитой GenCryptPrivateKey.exe, доступной в поставке connector со 138 версии Абонемент.

В утилите в поле PrivateKey необходимо прописать значение ключа прописанного в одноимённом параметре в connector.ini. После нажатия на кнопку "Сформировать" в поле CryptKey появится зашифрованная версия ключа, которую необходимо будет указывать в мобильном приложении в параметрах подключения к API.

Проводные контролеры - изменение параметров и прошивка

Изменение параметров игрового контролера GK

1. Изменение адреса на Ethernet-контролере.

По умолчанию адрес на контролере является 2001.

- Запустить утилиту gkTools.
- Нажать кнопку **Start**.
- В закладке install указать два блока цифр GK адреса, прошиваемого контролера (2001, 2002, 2003 и т.д.), нажать кнопку **set** и **restart**. После этого внизу появится сообщение connected with host ***
- Затем считать данные с точки доступа при помощи кнопки с **зелеными** стрелочками. В окне GK-address указать новые два блока цифр, нажать кнопку **без названия** (справа от поля GK address) и кнопку сохранить (**дискетка**).
- Нажать кнопку **restart**. Должна появиться надпись disconnected with host ***
- GK-адрес на контролере изменен.

В локальной сети не должно быть устройств с одинаковыми GK-адресами!

2. Для заливки файла прошивки firmware (формат *.fw) на игровом контролере (используется для обновления прошивок или изменения функционала контролера) понадобится утилита gkTools.

По умолчанию GK-адрес на контролере является 2001.

Процесс прошивки:

a. Запустить утилиту gkTools

b. Нажать кнопку Start.

с. В закладке install указать два блока цифр GK адреса, прошиваемого контролера (2001, 2002, 2003 и т.д.), нажать кнопку set и restart. После этого внизу появится сообщение connected with host ***

d. Зайти на вкладку Uploading и выбрать файл прошивки с расширением *.fw и нажать upload.

e. Нажать кнопку RestartAppFW и кнопку Restart. После этого внизу появится сообщение disconnected with host ***

Игровой контролер прошит.

Описание контроллера

Контроллер - это техническое устройство для управления оборудованием.

Контроллер имеет:

- **Два реле**, контакты которых рассчитаны на коммутацию тока до 3 А и 30 В постоянного тока.
- **Четыре логических ввода**.
При подключении необходимо соблюдать полярность входных напряжений. Напряжение логического нуля должно быть не более 1 В. Напряжение логической единицы не более 30 В.
- **интерфейс RS232C**, для подключения считывателя
- **интерфейс RS232C (RJ12)**, для подключения второго считывателя через переходник
- **интерфейс UART (RJ45)**, подключения платы расширения
- **интерфейс для управления жидкокристаллическим дисплеем**
- **интерфейс Ethernet**, для обмена данными с персональным компьютером.

Питание котроллера 12V 0.3A.

Внешний вид и размеры

<div>image not found or type unknown</div> <div>image not found or type unknown</div>	<div>image not found or type unknown</div>
Контроллер со снятой верхней крышкой	
<div>image not found or type unknown</div>	
Контроллер схематически отображены разъемы	
<div>image not found or type unknown</div>	

Функциональное назначение проводов и разъемов Основной логический провод

Таблица 1

=====	=====	=====	=====
		=====	=====
		=====	=====
		===	=====
Питание	+12V	Питание +12В / 0.3А	Питание контроллера
	GND	Питание +12В / 0.3А	
Реле №1	HP 1	Нормально разомкнутый РЕЛЕ1	Провода для формирования разрешающего сигнала на ВХОД
	СК 1	Средний контакт РЕЛЕ1	

=====	=====	=====	=====
НЗ 1	Нормально замкнутый контакт РЕЛЕ1		
Реле №2	НР 2	Нормально разомкнутый РЕЛЕ2	Провода для формирования разрешающего сигнала на ВЫХОД
	СК 2	Средний контакт РЕЛЕ2	
	НЗ 2	Нормальный замкнутый РЕЛЕ2	
Логический 1	D1+	Вывод "+" логического 1	Провода, фиксирующие факт прохода на "ВХОД". Работают совместно с "Логический 2"
	D1-	Вывод "-" логического 1	
Логический 2	D2+	Вывод "+" логического 2	Провода, фиксирующие факт прохода на "ВЫХОД". Работают совместно с "Логический 1"
	D2-	Вывод "-" логического 2	
Логический 3	D3+	Вывод "+" логического 3	Не используются в Абонемент
	D3-	Вывод "-" логического 3	
Логический 4	D4+	Вывод "+" логического 4	Не используются в Абонемент
	D4-	Вывод "-" логического 4	

Интерфейс UART

Выведен на восьми контактную телефонную розетку типа TJ4-8P8C.

Таблица №2


Изображение	Номер разъема TJ4-8P8C	Назначение вывода
-----	-----	-----
<div>  <div>image not found or type unknown</div> </div>	1	Вход данных считывателя
	2	Общий
	3	Выход данных на считыватель
	4	Общий

5	+5V
6	Общий
7	+5V
8	Общий

Интерфейс RS232C

Выведен на шести-контактную телефонную розетку типа TJ4-6P6C.
С этой розеткой применяется вилка TP-6P6C

Таблица №3

Изображение	Номер разъема TJ4-6P6C	Назначение вывода
-----	-----	----- -----
	1	+5V
	2	CTS
	3	Общий
	4	DTR
	5	RxD контроллера
	6	TxD контроллера

....