

Руководство по серверному приложению FEIG SERVER для работы с устройствами ISBC RFID



Оглавление

RE	ведені	ие	3
1.	Об	мен данными	5
2.	EPO	С-метки	6
3.	RFI	D-считыватели	7
4.	Уст	гановка и настройка сервера в OC Windows	8
5.	Уст	гановка и настройка сервера в ОС Linux	11
6.	Заг	туск сервера	14
7.	Ког	манды сервера	15
	7.1.	/addNewReader	15
	7.2.	/getReaders	15
	7.3.	/checkAnt	17
	7.4.	/switchAnt	17
	7.5.	/powerRF	19
	7.6.	/readOnce	20
	7.7.	/readStart	22
	7.8.	/readStop/	22
	7.9.	/readBuffer	22

Введение

ISBC RFID является одним из направлений деятельности группы компаний ISBC. Мы осуществляем поставки RFID-оборудования и техническое сопровождение интеграционных проектов, построенных на технологиях радиочастотной идентификации. ISBC заботится о том, чтобы для покупателей RFID-оборудования процесс интеграции в уже существующие системы учёта был максимально прост.

Серверное приложение NODE-RED для работы с устройствами ISBC RFID (далее Приложение) написано на языке *node-red* и является полностью открытым. Вы можете использовать данное приложение как есть, а также вы можете модифицировать его и дорабатывать под Ваши конкретные нужды. Отметим, что приложение является WEB-сервером, доступ к нему осуществляется через WEB браузер или посредством http-запросов. Структурная схема работы Приложения изображена на рисунке 1.

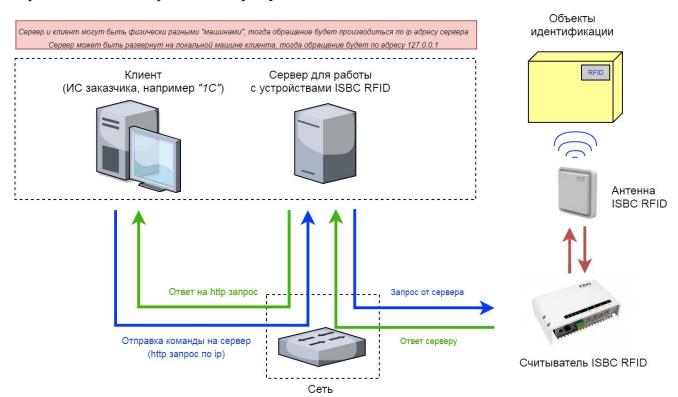


Рисунок 1. Структурная схема работы приложения

Доступ к приложению открыт на стандартном порту 1880, обращение имеет следующий вид <a href="http://<ip-cepвера>:1880">http://<ip-cepвера>:1880. Обратите внимание, что если на Вашем ПК или на машине, где установлено приложение установлены файрволы, антивирусы или другие приложения необходимо внести Приложение в «доверенные».



г. Зеленоград, ул. Новокрюковская, д. 3-Б, ГСК «Комета», Teл.: +7 (495) 133 00-01 E-mail: sale@isbc-rfid.ru

На момент публикации данного документа серверное приложение поддерживает следующие типы устройств:

- FEIG ELECTRONIC ID ISC.LRU1002
- FEIG ELECTRONIC ID ISC.MRU102

Подключение к вышеперечисленному оборудованию осуществляется по Ethernet. Наша компания планирует увеличить список поддерживаемых устройств *FIEG Electronic*. Следите за выходом новых версий.



г. Зеленоград, ул. Новокрюковская, д. 3-Б, ГСК «Комета», Тел.: +7 (495) 133 00-01 E-mail: sale@isbc-rfid.ru

1. Обмен данными

Существует несколько подходов к обмену данными (идентификаторы меток, время считывания и иные сведения) между RFID-считывателями и программой, а именно:

- 1. Обмен данными с помощью http-запросов;
- 2. Обмен данными с помощью загрузки из файла;
- 3. Обмен данными с помощью специальных драйверов.

В настоящем Руководстве будет описан только первый подход, так как серверное приложение использует именно его.

Если Вы не нашли ответ на интересующий Вас вопрос, то вы можете связаться со службой технической поддержки по электронной почте: support@isbc-rfid.ru.

Контактная информация ISBC RFID:

ООО "Интеллектуальные системы управления бизнесом"

Адрес:	г.Москва. г. Зеленоград, ул. Новокрюковская, д. 3-Б, ГСК «Комета»
Телефон:	+7 (495) 133 00-01
E-mail:	sale@isbc-rfid.ru
Сайт:	http://www.isbc-rfid.ru

2. ЕРС-метки

RFID-метка - это метка радиочастотной идентификации, которая состоит из интегральной схемы (чип), отвечающей за обработку и хранение информации, и антенны, которая принимает и передает сигнал.

RFID-метка имеет свой уникальный идентификатор, присваиваемый на этапе производства и обеспечивающий высокий уровень защиты изделия от подделок.

Идентификатор — это число в шестнадцатеричной системе счисления.



Рисунок 2. Таблица меток с номерами ЕРС

В систему учёта передаётся как минимум идентификатор RFID-метки. Но возможна так же передача времени считывания, номера антенны и иных сведений.

Примеры ЕРС-меток:

Метка для прачечных Flexible UHF Transponder "NOVO"

Тонкие мягкие метки легко пришиваются к любому текстильному изделию. Гарантированно выдерживают нагрузку более 200 циклов стирок при воздействии химических веществ, нагрева и перепада температур.



Рисунок 3. Пример метки для прачечных

Метка для крепежа на закругленных металлических поверхностях InLine TagTM Ultra Curve UHF

Хорошо зарекомендовала себя в использовании на возвратной металлической таре такой, как газовые баллоны, бочки для перевозки жидкостей.



Рисунок 4. Метка для крепления на металл

Компания ISBC предоставляет сотни типов RFID-меток, с полным перечнем которых можно ознакомиться на сайте http://www.isbc-rfid.ru/.

3. RFID-считыватели

Считыватели FEIG (около 30 моделей).

Продуктовая линейка FEIG Electronic состоит из нескольких типов RFID-считывтелей, разделенных по следующим группам:

- OBID i-scan UHF, IDENTIFICATION: RFID-считыватели UHF диапазона частот (860-960 МГц) малой, средней и большой дальности. Основные стандарты работы устройств это EPC, ISO 18000-6;
- OBID i-scan HF, IDENTIFICATION: RFID-считыватели HF 13,56 МГц малой, средней и большой дальности. Основные стандарты работы устройств это ISO15693, ISO 18000-3;
- Classic-Pro / CPR: настольные и настенные считыватели 13,56 МГц для организации физического доступа (СКУД) и для кодирования RFID-карт. Основные стандарты ISO 14443-A/B

Например, FEIG MRU102-PoE-LED UHF

Компактный размер и высокая производительность. MRU102-PoE-LED являются представителями семейства считываетелей FEIG OBID i-scan® UHF средней дальности.



Рисунок 5. Считыватель FEIG MRU102-POE-LED UHF

Ручной терминал для сбора данных



Рисунок 6. Ручной терминал сбора данных

Считыватель ISBC-RFID Tablet PC

Это промышленный защищенный планшет для использования в тяжелых условиях.

Планшет обладает высоким уровнем защиты от пыли и влаги - IP67, способен выдержать падения с высоты 1.2 метра



Pucyнoк 7. ISBC-RFID Tablet PC

4. Установка и настройка сервера в ОС Windows

Перед первым запуском приложения необходимо установить на пользовательский компьютер необходимые компоненты:

- 1) среду nodeJS и менеджер пакетов npm;
- 2) cpeдy node-red.

Примечание: для установки всех необходимых компонентов потребуется наличие подключения к сети интернет.

1) <u>Установка nodeJS и менеджера пакетов прт</u>

Скачиваем установочный файл nodeJS с официального сайта

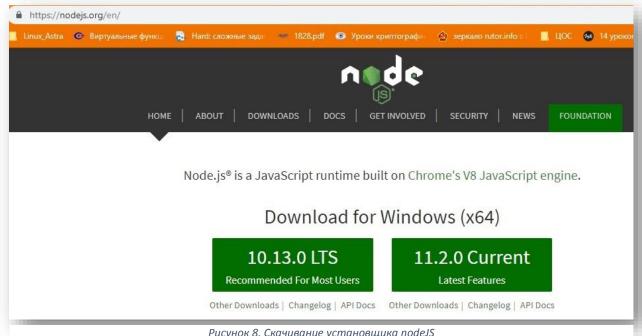


Рисунок 8. Скачивание установщика nodeJS

(https://nodejs.org/en/). Версия программы выбирается пользователем.

После того, как файл установки nodeJS полностью загрузился, запускаем его. Следуем всем инструкциям, какие будут возникать процессе установки.

Как только nodeJS успешно установился нажимаем кнопку "Finish". Окно установки программы закроется.



Рисунок 9. Завершение установки nodeJS

Открываем командную строку операционной системы и вводим следующие команды (после каждой команды нажимаем клавишу "Enter"):

- node --version
- npm --version

Каждая из этих команд проверяет и выводит в консоль версию установленного пакета. Они необходимы, чтобы убедиться в успешной установке nodeJS и менеджера пакетов npm.

После этого командную строку не закрываем, она нам еще потребуется. Переходим к пункту 2.

2) <u>Установка node-red</u>

В открытом окне командной строки вводим $npm\ install\ -g\ --unsafe\ -perm\ node\ red\$ и нажимаем клавишу "Enter". Запустится процесс скачивания пакетов из интернета и их установка на локальный компьютер. Во время установки на экране будут появляться сообщения с предупреждениями (WARN) - игнорируем их.

```
C:\Users\BloodLicch>npm install -g --unsafe-perm node-red

npm WARN
/status/
/status/
npm WARN
deprecated mailparser@0.6.2: Mailparser versions older than v2.3.0 are deprecated

npm WARN
npm WARN
npm WARN
deprecated mimelib@0.3.1: This project is unmaintained

npm WARN
npm
```

Рисунок 10. Установка node-red

Примерно через минуту node-red установится и на экране появится сообщение о количестве установленных пакетов.

```
C:\Users\BloodLicch\npm install -g --unsafe-perm node-red
npm WARN deprecated nodemailer@1.11.0: All versions below 4.0.1 of Nodemailer are deprecated. S
/status/
npm WARN deprecated mailparser@0.6.2: Mailparser versions older than v2.3.0 are deprecated
npm WARN deprecated mimelib@0.3.1: This project is unmaintained
npm WARN deprecated mailcomposer@2.1.0: This project is unmaintained
npm WARN deprecated buildmail@2.0.0: This project is unmaintained
C:\Users\BloodLicch\AppData\Roaming\npm\node-red-pi -> C:\Users\BloodLicch\AppData\Roaming\npm\
\node-red-pi
C:\Users\BloodLicch\AppData\Roaming\npm\node-red -> C:\Users\BloodLicch\AppData\Roaming\npm\nod
> bcrypt@2.0.1 install C:\Users\BloodLicch\AppData\Roaming\npm\node_modules\node-red\node_modul
> node-pre-gyp install --fallback-to-build

[bcrypt] Success: "C:\Users\BloodLicch\AppData\Roaming\npm\node_modules\node-red\node_modules\b
lib.node" is installed via remote
+ node-red@0.19.5
added 388 packages from 342 contributors in 70.445s
```

Рисунок 11. Завершение установки node-red

После того как все установилось переходим в папку с установленным nodered ($C:\users\< Mms_nonb3oвameлs>\.node-red$) и копируем в нее файл Server.json.

	Тип: JSON File Размер: 68,6 КБ Дата изменения: 18.11.2018 19:13		
🏂 settings.js	18.11.2018 19:15	файл JavaScript	12 KB
Server.json	<u>18,11,2018 19:13</u>	JSON File	<u>69 KB</u>
🖵 package.json	18.11.2018 19:12	JSON File	1 KB
.config.json.backup	18.11.2018 19:12	Файл "BACKUP"	12 KB
🎜 .config.json	18.11.2018 19:16	JSON File	12 KB
node_modules	18.11.2018 19:12	Папка с файлами	
lib	18.11.2018 19:12	Папка с файлами	

Далее открываем файл *settings.js* и добавляем строчку *net:require('net')* в блок *functionGlobalContext* как показано на изображении ниже.

```
// can be accessed in a function block as:
// context.global.os

functionGlobalContext: {
    net:require('net')
    // os:require('os'),
    // jfive:require("johnny-five"),
    // j5board:require("johnny-five").Board({repl:false})
},

// Context Storage
```

Рисунок 12. Правка файла settings.js

Сохраняем изменения. Все необходимые пакеты установлены и сервер готов к работе.

5. Установка и настройка сервера в ОС Linux

В ОС Linux прежде чем запускать сервер необходимо установить дополнительные пакеты и компоненты. Для этого необходимо обновить список доступных пакетов и установить пакет с названием *nodered*. Для этого открываем командную строку (terminal) и вводим следующие команды:

1) sudo apt-get update

```
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://raspbian.raspberrypi.org/raspbian stretch InRelease
Get:2 http://archive.raspberrypi.org/debian stretch/main armhf
Get:3 http://raspbian.raspberrypi.org/raspbian stretch/main armhf
MB]
Get:4 http://archive.raspberrypi.org/debian stretch/main armhf Pac
Get:5 http://archive.raspberrypi.org/debian stretch/ui armhf Pac
Get:6 http://raspbian.raspberrypi.org/raspbian stretch/non-free a
95.5 kB]
Fetched 12.0 MB in 19s (624 kB/s)
Reading package lists... Done
```

Рисунок 13. Результат команды sudo apt-get update

2) sudo apt-get install nodered

```
pi@raspberrypi:~ $ sudo apt-get install nodered
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libhttp-parser2.8 libuv1 nodejs nodejs-doc
The following NEW packages will be installed:
  libc-ares2 libhttp-parser2.8 libuv1 nodejs nodejs-doc nodered
0 upgraded, 6 newly installed, 0 to remove and 71 not upgraded.
Need to get 10.4 MB of archives.
After this operation, 83.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

Рисунок 14. Запуск установки node-red (команда sudo apt-get install nodered)

Примечание: в зависимости от версии Linux содержание сообщения после команд выше могут отличаться.

Вводим у и нажимаем клавишу "Enter". Начнётся выгрузка необходимых пакетов из сети интернет и их установка в систему. После того как все установится в командной строке появится сообщение, что пакет успешно установлен в систему.

```
Unpacking nodejs (8.11.1~dfsg-2~bpo9+1) ...
Selecting previously unselected package nodejs-doc.
Preparing to unpack .../4-nodejs-doc_8.11.1~dfsg-2~bpo9+1_all.deb ...
Unpacking nodejs-doc (8.11.1~dfsg-2~bpo9+1) ...
Selecting previously unselected package nodered.
Preparing to unpack .../5-nodered_0.19.4_armhf.deb ...
Unpacking nodered (0.19.4) ...
Setting up nodejs-doc (8.11.1~dfsg-2~bpo9+1) ...
Processing triggers for mime-support (3.60) ...
Processing triggers for desktop-file-utils (0.23-1) ...
Setting up libuv1:armhf (1.18.0-3~bpo9+1) ...
Processing triggers for libc-bin (2.24-11+deb9u3) ...
Processing triggers for man-db (2.7.6.1-2) .
Processing triggers for gnome-menus (3.13.3-9) ...
Processing triggers for hicolor-icon-theme (0.15-1)
Setting up libc-ares2:armhf (1.14.0-1~bpo9+1) ...
Setting up libhttp-parser2.8:armhf (2.8.1-1-bpo9+1) ...
Setting up nodejs (8.11.1~dfsg-2~bpo9+1) ...
update-alternatives: using /usr/bin/nodejs to provide /usr/bin/js (js) in au
ode
Setting up nodered (0.19.4) ...
Processing triggers for libc-bin (2.24-11+deb9u3)
```

Рисунок 15. Завершение установки node-red

После того как все установилось, переходим в папку с установленным *node-red* (*.node-red*) и копируем в нее файл *Server.json*.

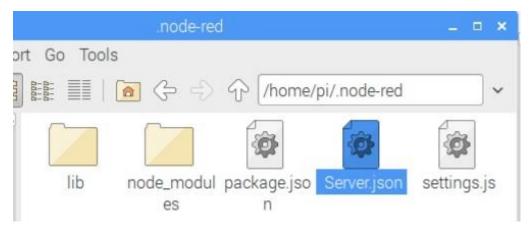


Рисунок 16. Копирование файла сервера в папку node-red

Далее открываем файл settings.js и добавляем строчку net.require('net') в блок functionGlobalContext как показано на изображении ниже.



```
// Anything in this hash is globally available to all
// It is accessed as context.global.
// eg:
// functionGlobalContext: { os:require('os') }
// can be accessed in a function block as:
// context.global.os

functionGlobalContext: {
    net:require('net')
    // os:require('os'),
    // jfive:require("johnny-five"),
    // j5board:require("johnny-five").Board({repl:fal}),
// Context Storage
```

Рисунок 17. Правка файла settings.js

Сохраняем изменения. Все необходимые пакеты установлены и сервер готов к работе.

6. Запуск сервера

Запуск сервера для работы с устройствами FIEG осуществляется из командной строки. Для этого:

1. В Вашей операционной системе необходимо открыть консоль (в ОС Window: $win+R \rightarrow в$ открывшемся окошке ввести cmd и нажать клавишу Enter. В Linux открыть наиболее удобный Terminal);

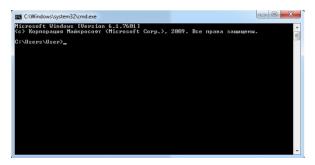


Рисунок 18. открытие консоли в ОС Windows

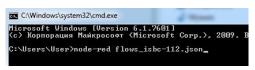


Рисунок 19. Запуск сервера

2. В открывшемся окне (в зависимости от вашей ОС вид данного окна может отличаться) ввести команду node-red < имя файла сервера>.json;

S node-red

If Now 15:15:41 - [warm]

If Now 15:15:41 - [info] Settings file: "Wisers\mukailov\.node-red\settings file is wisers\mukailov\.node-red\settings file is wisers\mukailov\.node-red\mukailov\.node-red\settings file is wisers\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.node-red\mukailov\.n

our settings file. Node-RED will then re-encrypt your credentials ile using your chosen key the next time you deploy a change.

РИСУНОК 20. ЛОГ СОБЫТИЙ СЕРВЕРА ПОСЛЕ ЕГО ЗАПУСКА

ated key is lost for any reason, your credentials coverable, you will have to delete it and re-enter

3. Убедиться, что сервер запустился.

После выполнения описанной выше процедуры запустится сервер для работы с устройствами FIEG. По умолчанию, за сервером закрепляется порт 1880. Для подключения к нему (отправки команд)



Рисунок 21. ІР-адреса сервера

необходимо ввести ip-адрес сервера: порт.

Таким образом, если Вы собирается

подключиться к серверу, запущенному на Вашем компьютере, необходимо ввести в строке браузера $127.0.0.1:1880/<\kappa$ оманда>. Если сервер находится в сети (в одной сети с компьютером, с которого будут отправляться команды), то вместо 127.0.0.1 необходимо ввести сетевой адрес устройства с запущенным сервером (например, 192.168.10.12).

7. Команды сервера

После того, как Вы успешно запустили сервер, ему можно посылать команды для работы с устройствами FIEG. Перед началом отправки команд настоятельно просим убедиться в том, что устройство FIEG находится в одной сети с сервером, на него подано питание и подключены антенны.

7.1. /addNewReader

Данная команда необходима для установки параметров подключения к считывателю, а именно *vendor* (*mun устройства*), *connection type* (*mun подключения*), *ip-адрес* устройства и *порт* и должна вызываться перед всеми прочими командами.

Команда имеет следующий вид:

```
<ip cepsepa>:1880/addNewReader?vendor=<mun ycmpoйcmвa>&
connectionType=<mun nodключения>&host=<ip-ycmpoйcmвa>&
port=<nopm ycmpoйcmвa>
```

<u>Примечание</u>: данное приложение работает только с устройствами FEIG, которые поддерживают подключение по TCP.

Например, у Вас имеется считыватель FEIG, поддерживающий подключение по TCP, с ір 192.168.10.10 и портом 1001. Вы хотите передать данные параметры в сервер для дальнейшего подключения к считывателю.

Параметры, выделенные желтым цветом, вводятся пользователем и зависят от его параметров сети. В случае успеха в браузере должно отобразиться сообщение о том, что адрес устройства FIEG успешно установлен, либо выдать ошибку (например, если не указать порт считывателя).



Рисунок 22

7.2. /getReaders

Данная команда не имеет параметров и позволяет узнать версию прошивки устройств, его тип, уникальный номер (в шестнадцатеричном виде) и тип подключения.

Команда имеет следующий вид:

```
<ip сервера>:1880/getReaders
```

После нажатия клавиши *Enter* сервер начнет обмениваться командами с устройством FIEG по параметрам, установленным в команде /setTCP (см. Команды сервера пункт 1). После успешного выполнения команда возвращает версию прошивки, тип устройства и его уникальный номер в JSON-объекте следующего вида:

```
("readers":[{"reader":"ID ISC.LRU1002", "readerID":"17ddfa6e", "firmware":"1.1.0", "connectionType":"TCP", "vendor": "FEIG"}]}
```

Рисунок 23

Вместо текста, выделенного желтым цветом, возвращаются данные по конкретному устройству.

Если сервер не смог связаться с устройством, или по каким-то другим причинам обмен данным с устройством нарушился, то вернется ошибка в виде JSON-объекта (примечание: все последующие команды имеют аналогичное поведение).

```
("errno":"ETIMEDOUT","code":"ETIMEDOUT","syscall":"connect","address":"192.168.10.10","port":10001}
```

Рисунок 24

После успешного выполнения данной команды можно посылать на сервер другие запросы, описанные ниже.

7.3. /checkAnt

<u>Примечание</u>: данная команда доступна только для устройств FIEG серии LRU.

Команда /checkAnt не имеет параметров и предназначается для определения физического подключения антенн к устройствам LRU. После подачи данной команды на сервер, начинается процесс опроса считывателя серии LRU на наличие физического соединения с антеннами. В процессе общения считыватель будет перезагружаться, что занимает время в районе 10 секунд. В связи с этим, ответ данную команду придет только после того, как устройство полностью завершит процесс перезапуска.

Команда имеет следующий вид

<mark><ip сервера></mark>:1880/checkAnt

В результате выполнения команды сервер возвращает JSON-объект, в котором описывается текущее состояние каждой антенны (connect-подключена и disconnect-отключена).



Рисунок 25

7.4. /switchAnt

Данная команда предназначена для управления состоянием антенн и реализована в двух вариантах: без параметров и с параметрами.

a) /switchAnt без параметров

В данном варианте команде имеет следующий вид:

<mark><iр сервера></mark>:1880/checkAnt

В результате успешного выполнения команды сервер возвращает JSON-объект, содержащий состояние для каждой антенны подключенного устройства (оп-включена или off-выключена).



Рисунок 26



г. Зеленоград, ул. Новокрюковская, д. 3-Б, ГСК «Комета», Тел.: +7 (495) 133 00-01 E-mail: sale@isbc-rfid.ru

Примечание: состояние *on* (антенна включена) сообщает только о том, что выход антенны от устройства подаются сигналы. Однако это не означает, что антенна имеет физическое подключение к устройству в момент опроса.

b) /switchAnt с параметрами

В данном варианте в команде передаются номера антенн, которые будут включены на подключенном устройстве. Все прочие антенны (не указанные в параметрах команды), будут выключены.

Команда имеет следующий вид:

<ip сервера>:1880/checkAnt?Ant=<номера включаемых антенн >...

В случае успешного выполнения север присылает сообщение "success".



Рисунок 27

Примечание: диапазон указываемых антенн для включения от 1 до 4. Все прочие номера антенн приниматься не будут. Таким образом, если подать команду с номерами антенн большими 4, то к сообщению об успешном выполнении добавится оповещение о том, что некоторые номера антенн не приняты командой.

В данной команде видно, что в параметрах указаны номера антенн от 1 до 8. В результате, на устройстве включились антенны с номерами 1,2,3 и 4, а про остальные номера (от 5 до 8) выведено сообщение "Wrong numbers".



Рисунок 28

В примере ниже отправляется команда, у которой в параметрах перечислены только неверные номера антенн. В таком случает от сервера вернется сообщение о том, что включена антенна по умолчанию (для LRU



г. Зеленоград, ул. Новокрюковская, д. 3-Б, ГСК «Комета», Тел.: +7 (495) 133 00-01 E-mail: sale@isbc-rfid.ru

– 1я, для MRU – внутренняя), а также сообщение "Wrong numbers" с непринятыми номерами антенн.

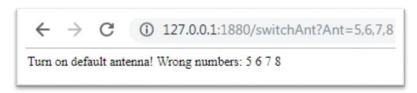


Рисунок 29

7.5. /powerRF

Данная команда предназначена для управления состоянием антенн и реализована в двух вариантах: без параметров и с параметрами.

a) /powerRf без параметров

В данном варианте команда имеет следующий вид:

<mark><ip сервера></mark>:1880/powerRF

В результате успешного выполнения команды сервер возвращает JSON-объект, содержащий выходную мощность в ваттах (W) для каждой антенны подключенного устройства.



Рисунок 30

Примечание: значение мощности вернется абсолютно для всех антенн устройства, как включенных, так и выключенных.

b) /powerRf с параметрами

Установка одинаковой мощности на всех антеннах

Данный вариант команды доступен для всех типов устройств с которыми сервер умеет обмениваться данными (т.е. для LRU и MRU). Для установки мощности в команду добавляется параметр *Ants* и после знака "=" указывается желаемое значение выходной мощности считывателя.

В данном варианте команда имеет следующий вид:

<mark><ip сервера></mark>:1880/powerRF?Ants=<mark><мощность></mark>

В случае успешного выполнения сервер вернет сообщение "success".





г. Зеленоград, ул. Новокрюковская, д. 3-Б, ГСК «Комета», Тел.: +7 (495) 133 00-01 E-mail: sale@isbc-rfid.ru

Примечание: стоит учитывать тот факт, что для устройств LRU и MRU диапазон устанавливаемых значений мощности различен. Для MRU -0.05W, 0.1W, 0.2W, 0.3W, 0.4W и 0.5W. Для LRU диапазон значений гораздо шире: от 0.1W до 2W с шагом 0.1.

<u>Установка мощности на конкретных антеннах (только для LRU)</u>

Для установки мощности на конкретных антеннах указывается номер антенны и значение мощности в ваттах (образуются пары данных: *антенна – мощность*). Команда имеет следующий вид:

<ip ceрвера>:1880/powerRF?Ant<№ антенны>=<мощность>...

В случае успешного выполнения сервер вернет сообщение "success".



Рисунок 32

<u>Примечание</u>: на неуказанных в команде антеннах значение мощности не меняется.

7.6. /readOnce

Команда предназначена для разовой инвентаризации (разового чтения меток в зоне видимости антенн подключенного устройства) и реализована в двух вариантах: команда только с ID номером считывателя и с дополнительными параметрами.

<u>Примечание</u>: при вызове данной команды обязательно передается ID номер считывателя, подключенного к FEIG Server (параметр *readerID*).

а) /readOnce только с ID номером считывателя

При вызове команды *readOnce* без параметров происходит единоразовое считывание с включенных на устройстве антенн (включить/выключить антенны на устройстве, а также узнать какие антенны активны можно командой *switchAnt* из пункта 7.4 данного руководства).

Команда имеет следующий вид:

<ip сервера>:1880/readOnce&readerID=<id считывателя>

В случае обнаружения меток в зоне видимости антенн, команда возвращает JSON-объект, в котором находятся EPC и время (в стандарте ISO 8601) чтения каждой конкретной метки.



Рисунок 33

Если меток в зоне видимости антенн не обнаружено, вернется сообщение "err".

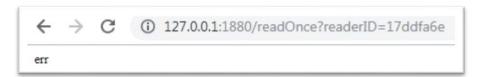


Рисунок 34

b) /readOnce с дополнительными параметрами

Данный вариант команды имеет аналогичное поведение, что и ее вариант без параметров, за тем исключением, что считывание производится с антенны, которая указаны в параметрах команды.

Команда имеет следующий вид:

<mark><ip сервера></mark>:1880//readOnce&readerID=<mark><id считывателя></mark>&Ant=<mark><номер антенны></mark>

Например, чтение производится только с 3й антенны:

```
← → C ① 127.0.0.1:1880/readOnce?readerID=17ddfa6e&Ant=3

{"tags":[{"EPC":"0101010019566FFFFF001257","time":"2018-11-16713:40:01"},{"EPC":"0101010022735FFFFF000001","time":"2018-11-16713:40:01"}]}
```

Рисунок 35

Например, чтение производится с 1й и 3й антенны:



Рисунок 36

7.7. /readStart

Данная команда включает буферный режим работы устройств FIEG и самого сервера. Таким образом, после выполнения данной команды в памяти сервера будет производиться накопление базы обнаруженных меток (чтобы считать базу воспользуйтесь командой *readBuffer* из п. 7.9 данного руководства).

Команда имеет следующий вид:

<mark><iр сервера></mark>:1880/readStart

В случае успеха команда возвращает "success":



Рисунок 37

7.8. /readStop

Команда предназначена для остановки буферного режима чтения и обнаружения меток в зоне видимости антенн устройств FIEG.

Команда имеет следующий вид:

<mark><ip сервера></mark>:1880/readStop

В случае успеха команда возвращает "success"

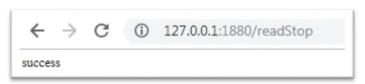


Рисунок 38

7.9. /readBuffer

Команда предназначена для чтения базы меток из памяти сервера и имеет следующий вид:

<mark><ip сервера></mark>:1880/readBuffer

В результате возвращается JSON-объект со списком меток и временем считывания каждой конкретной метки. Формат возвращаемого JSON-объекта представлен ниже:

```
{
    "tags":[
    {
        "epc": "epc номер метки",
        "time": "время чтения по стандарту ISO"
    },
    ...
]
```

Внутренняя база сервера очищается.

```
← → C (i) 127.0.0.1:1880/readBuffer

{"tags":[{"EPC":"9101010019566FFFF001257","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6BF9","time":"2018-11-16T14
{"EPC":"300ED89F3350007FC47E6B95","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B50","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6C45","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6BC6","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6BC6","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007FC47E6B70","time":"2018-11-16T14:18:51"},{"EPC":"300ED89F3350007F
```

Рисунок 39

Если в базе сервера меток нет, то выводится сообщение "No data in reader buffer memory".



Рисунок 40