

WEBSDR

Наверное, всем известен популярный сервис в интернете, веб-сдр приёмник: **WEBSDR.ORG**

На сегодняшний день этот сервис включает 169 официальных приёмников по всему миру и неизвестное количество не зарегистрированных приёмников. Обусловлено это тем, что автор этого, поистине бесценного произведения искусства PA3FWM, Питер-Тьерк де Бур, являющийся доцентом голландского университета Твенте сильно ограничил распространение ядра приёмника. Требования Питера понятны. Запуская сервер веб-сдр на своём компьютере, было бы крайне желательно обеспечить его бесперебойную работу для большого количества подключающихся пользователей (клиентов) со всего мира. Для этого необходимо было иметь мощный компьютер и довольно быстрый интернет. Этот принцип заложен был в изначальное требование к тому, кому Питер раздавал приёмники. Напомню, эти требования были актуальны в конце 00-ых, начале 10-ых годов, когда похвастаться подобными ресурсами могли не все. Но время идёт. На сегодняшний день, компьютер, который может с лёгкостью обрабатывать хоть несколько SDR-приёмников и иметь широкополосный скоростной интернет в больших городах (и даже в сёлах) если есть не у каждого радиолюбителя, то у каждого третьего – это точно. К тому-же, как говорят слухи, автор уже как года 2 перестал поддерживать этот проект и не обновляет его.

Именно по данной причине, я решил написать статью для всех интересующихся данной тематикой и описать в подробностях, как работает этот сервис.

КАК РАБОТАЕТ СЕРВЕР WEBSDR.

Как именно работает сам сервер практически никому не известно, т.к. автор все исходные коды сервера держит всё это время в секрете, чем сильно ограничивает его развитие. Для людей, разбирающихся глубоко в ЦОС, наверное, нет секретов, но для большинства это и не важно. Важно понимание самого принципа работы всей системы.

Сервер собой представляет скомпилированную т.н. сервер Java-машину, которая обрабатывает входящий IQ-поток данных от физического приёмника (или нескольких приёмников) и создаёт свой индивидуальный виртуальный приёмный канал с заданными свойствами (модуляция/полоса/громкость/индикация уровня сигнала) для каждого вновь подключившегося клиента + для всех имеется одинаковый водопад или спектр с оптимальными свойствами для низкоскоростных соединений к серверу извне. Для каждого клиента данные с приёмника упаковываются в TCP пакеты и отправляются на удалённый компьютер. На открытой странице приёмника эти данные обрабатываются. Вы видите панораму, управляете приёмником, слышите звук. При этом сетевой трафик получается минимальным, а нагрузка на компьютер, в отличии от реального SDR-приёмника – ничтожна!

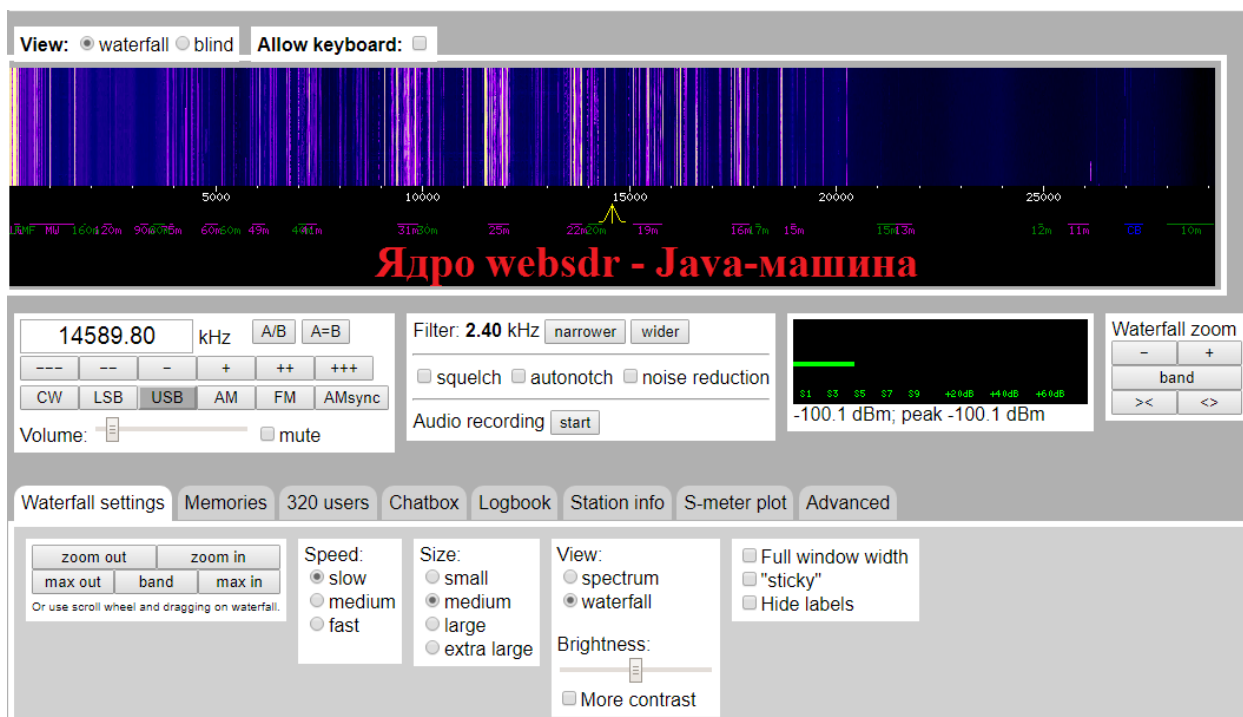
Надо отметить, что долгое время обработкой графики и звука делалось на Java, встроенном в интернет-браузер. Сегодня, это всё делается на базе новой технологии HTML-5. Обработчик вы можете выбирать сами на странице приёмника, но по соображениям безопасности, лучше применять именно HTML-5.

Сам сервер «крутится» на базе ОС Linux. Для этого подходит всё семейство ОС Ubuntu или Debian или их ответвления. Подходят даже самые минималистические образы систем. Лишь бы работало Ядро ОС и сетевое оборудование. Все настройки сервера и старт осуществляются из командной строки. Для удобства, конечно, можно использовать и ОС с графической оболочкой, но в самых крайних вариантах со слабым железом, описанных ниже желательно использовать минимальный консольный вариант ОС.

Как было сказано выше, сервер websdr-приёмника – это совсем крошечное ядро на Java. Всё остальное представление websdr-страницы является надстройкой к этой Java-машине и представлено в виде HTML-кода и JavaScript-кода. В этом коде описано, как страница выглядит, начальные настройки страницы, состояние приёмника и точки входа управления приёмником.

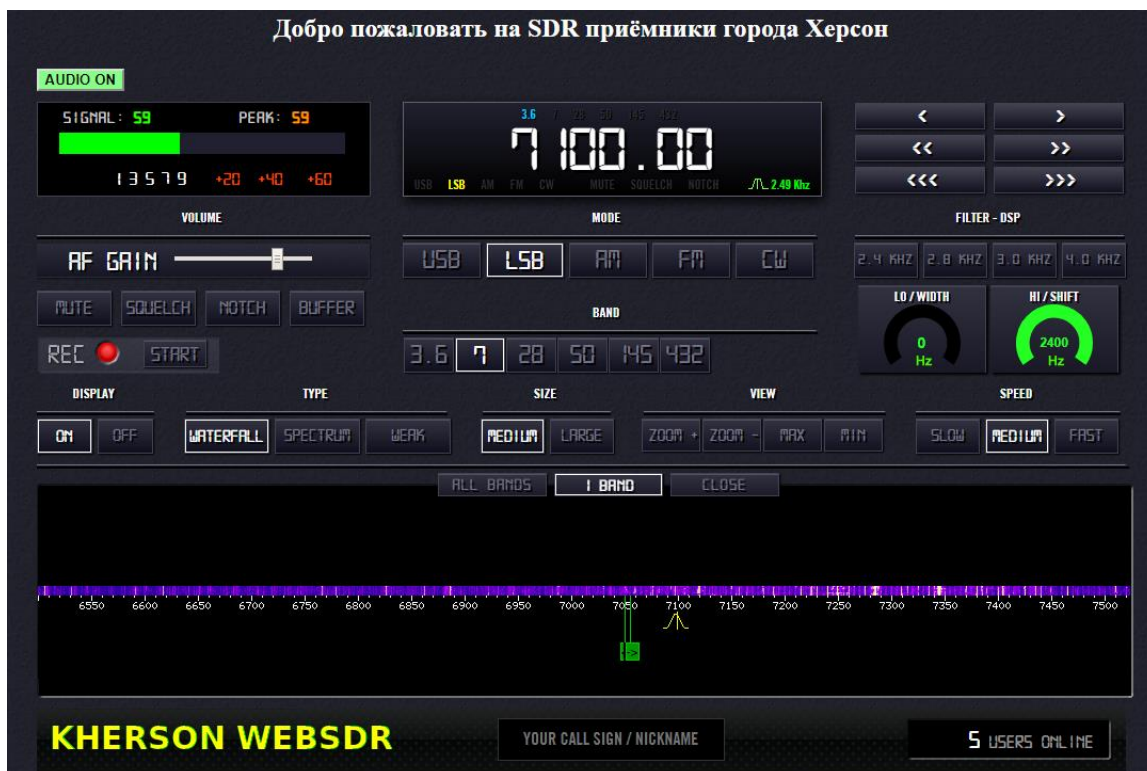
Кроме того, реализована возможность слушать websdr-приёмник на любом более-менее современном телефоне на базе Android или i-OS.

Изначально, автор максимально упростил оформление и долгое время оформление страниц у большинства приёмников было по авторский серым и минималистичным.



Но, когда среди радиолюбительского сообщества нашлись мастера веб-дизайна, некоторые страницы веб-приёмников преобразились до неузнаваемости и стали по-своему прекрасными!

Я приведу лишь пару примеров, что бы вы могли оценить возможности оформления при творческом подходе:



WebSDR receiver in Moscow, Russia (ex. websdr.tech)

ИНФОРМАЦИЯ



Зачем нужен
WebSDR-приемник?

| # | Name | Chip / Tuner | Band | Width | Antenna |
|----|---------|------------------|-------------------|-------|-----------------|
| 01 | 127 MHz | RTL-SDR / R820T2 | 126.956 — 129.004 | | Sirio T3-27 |
| 02 | 145 MHz | RTL-SDR / R820T2 | 143.956 — 146.004 | | Diamond X-50 |
| 03 | 446 MHz | RTL-SDR / R820T2 | 445.656 — 447.704 | | Nagoya UT-102UV |
| 04 | 131 MHz | RTL-SDR / R820T2 | 130.156 — 132.204 | | Diamond SG-M507 |

QTH Locator: K085UP The server was launched on February 1, 2017.
Support By WebSDR.Space Powered By WebSDR.Org



ВАЖНАЯ
информация

ПРИЕМНИК

PMF CW LSB USB AM FM FILTER 10.00 MHz

145500.00 5:23 24.0 dB

RECORD BAND 2M RX MUTE SQL NOTCH

Waterfall Spectrum Weak sigs Strong sigs

CW-W LSB-W USB-W AM-W FM-W

CW LSB USB AM FM

CW-N LSB-N USB-N AM-N FM-N

NARROWER WIDER

Band Zoom + Zoom - Max Zoom

GAIN MOD 1 2 3 4 5 NOISE FLOOR OVERLOAD

SIGNAL -120 -100 -80 -60 -40 -20

51-53-55-57-59 +20 +40 +60

Mute Squelch Autotune

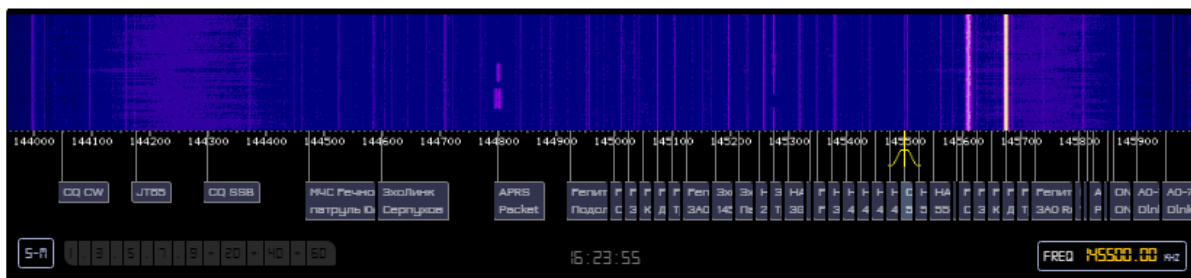
Volume RECORD

Band: 127 MHz 145 MHz 446 MHz 131 MHz

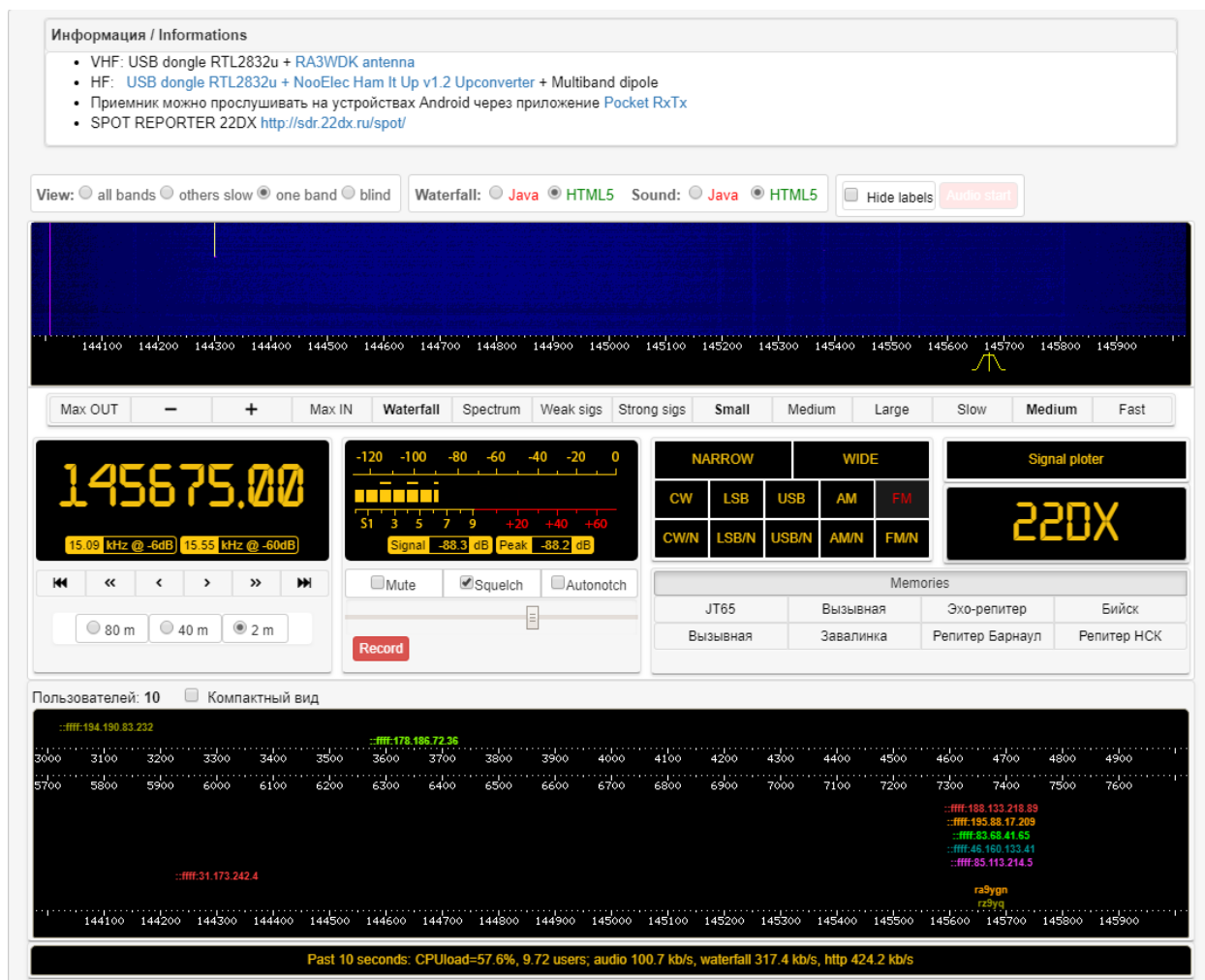
РАСШИРЕННОЕ УПРАВЛЕНИЕ

ВОДОПАД

Вид: all bands others slow one band blind Водопад: Java Html 5 Звук: Java Html 5



Band Zoom + Zoom - Max Zoom Shift Shift L Edge L Edge U Edge U Edge



ТРЕБОВАНИЯ И ВОЗМОЖНОСТИ В ОТНОШЕНИИ КОМПЬЮТЕРНОГО ЖЕЛЕЗА

На заре развития websdr-сервера в качестве физического приёмника закладывалось применение простых приёмников типа SoftRock, работающих со звуковой картой компьютера. На начальном этапе приёмники имели полосу 48/96/192кГц и поддерживали только КВ-участки. Когда в широкую продажу зашли из Китая экстра-дешёвые 8-битные SDR-свистки - это произвело переворот! Автор включил их в поддержку железа и сервер стал практически универсальным. Следует понимать, что сервер никак не связан с тем, какой диапазон вы хотите транслировать. Диапазон принимаемых частот зависит только от железа SDR-приёмника. Сервер занимается только обработкой IQ потока. В тоже время, автор зачем-то заложил ограничение в полосу принимаемого потока (чем в очередной раз очень сильно ограничил возможности и сферу применения сервера) до 2МГц. Сам он, на своём авторском сервере использует всю доступную ему полосу в 30МГц и делиться подобным решением, по отзывам категорически отказывается. Мы точно не знаем, но, наверное, на это у него есть веские причины, судить его за это не будем.

Для обеспечения бесперебойной работы для большого количества подключающихся пользователей (клиентов) со всего мира необходимо соблюсти 2 условия: наличие мощного компьютера и широкополосного скоростного интернета с «белым» IP-адресом. Если вы хотите сделать свой личный приватный сервер websdr или сервер для друзей, то требования к железу и интернету существенно упрощаются. Нагрузка на компьютер складывается из количества подключившихся к серверу клиентов и применяемого (применяемых) SDR-приёмников.

Рассмотрим следующие варианты реализации компьютерного железа:

- 1) Построение производительного websdr-сервера для общего применения на базе нескольких SDR-приёмников с полосой до 2МГц:
 - Компьютер с процессором не ниже i5 или i7 последних поколений с 8 или 16Гб памяти
 - Максимально скоростной, желательно оптический интернет с белым IP-адресом.

Такой сервер позволит подключаться к приёмнику неограниченному количеству клиентов со всего мира.

- 2) Построение среднего websdr-сервера для общего применения на базе нескольких SDR-приёмников с полосой до 2МГц:
 - Компьютер с процессором не ниже i3, i5 или i7 из 2...6 поколений с 8 или 16Гб памяти
 - Максимально скоростной LAN-интернет с белым IP-адресом.

Такой сервер позволит подключаться к приёмнику многим клиентам со всего мира в количестве 30...60 человек.

- 3) Построение слабого websdr-сервера для индивидуального применения или для ближайших друзей на базе нескольких или одного SDR-приёмника с полосой до 2МГц:
 - Компьютер с процессором не ниже 2-х ядерного Целерона или, лучше, i3/i5/i7 из 1 или 2 поколений i-core с 2, 4 или 8Гб памяти
 - Скоростной LAN-интернет 10...50МБс, желательно с белым IP-адресом, но можно и без белого IP. Для крайних случаев подойдёт модем с 4G интернетом.

Такой сервер позволит подключаться к приёмнику вам или вашим друзьям со всего мира в количестве 1...30 человек.

- 4) Построение простейшего websdr-сервера для индивидуального применения или для ближайших друзей на базе одного SDR-приёмника с полосой до 1МГц или пары SDR-приёмников полосой 96...256кГц:
 - Микрокомпьютер Raspberry Pi 3 или 4 в максимальной конфигурации
 - Скоростной LAN-интернет 10...50МБс, желательно с белым IP-адресом, но можно и без белого IP. Для крайних случаев подойдёт модем с 4G интернетом.

Такой сервер позволит подключаться к приёмнику вам или вашим друзьям со всего мира в количестве 1...10 человек.

Примечание 1: автором этих строк был опробован websdr-сервер на базе Raspberry Pi Zero. Он более-менее сносно позволял прослушивать на одном SDR-свистке УКВ-диапазон полосой 256кГц 10-и человекам. Так же, был опробован вариант сервера на базе нетбука HP x360 с процессором Celeron N2840. Он более-менее сносно позволял прослушивать на 3х SDR-свистках разные УКВ-диапазоны полосой по 1МГц 20-и человекам.

Примечание 2: Не рекомендуется применение компьютеров на базе процессоров поколения ниже i-core, т.е. всяких Core 2 Duo и, тем более, всяких Целеронов предыдущих поколений.

Поддерживаемые приёмники:

- 1) Любые классические SDR-приёмники, любого диапазона на базе применения звуковых карт. Главным критерием должна быть поддержка звуковой карты в Linux. Полоса сервера - до 196кГц.
- 2) SDR-приёмники на базе R820T(T2)+RTL2832. Полоса сервера - до 2МГц.
- 3) SDR-приёмник Afedri. Полоса сервера – до 768кГц.
- 4) SDR-приёмники на базе клонов SDRPlay. (Пока не опробовано, но пишут, что до полосы 2МГц тоже работает)

- 5) Любой другой SDR или DDC приёмник с IQ выводом данных в звуковом или TCP-виде.
(Тема отдельных исследований и экспериментов)

Т.о. вы можете легко собрать свой собственный сервер на базе дешёвых SDR-свистков для себя любимого как в пределах внутренней домашней (или не домашней, а на работе) сети или с выходом в «большой веб». Свой собственный сервер вы можете не светить в общей сети и/или на сайте-агрегаторе автора, достаточно правильно сконфигурировать сервер.

СОСТАВ СЕРВЕРА WEBSDR

Собственно, websdr-сервер – это один крохотный запускаемый файл, размером около 200кБ, который запускается из командной строки или специального скрипта автозапуска. К этому файлу прилагается пару файлов конфигурации предварительного состояния и оформления. Так же, в состав оформления сервера входит полноценная или мобильная версия HTML-страницы, открываемая клиентом при подключении к серверу.

Состав архива сервера состоит из следующих папок:

websdr/pub – папка содержит публичные файлы (спецификация для конкретного сайта)

websdr/cfg – папка содержит файлы конфигурации для конкретно вашего сервера.

websdr/dist – папка содержит запускаемые и служебные файлы сервера

websdr/dist/doc – папка содержит документацию.

websdr/dist/log – папка содержит подробные файлы журнала.

/home/websdr/dist/pub2 – папка содержит набор общедоступных файлов оформления вашего сайта web-sdr.

/home/websdr/dist/pub2/tmp – папка содержит временные публичные файлы.

Автор этой статьи потрудился перевести полностью документацию на русский язык. Выкладываю её на благо всего радиолюбительского сообщества. Перевод вы найдёте в приложении.

РАБОТА СЕРВЕРА И ОСОБЕННОСТИ

Как было сказано выше, websdr-сервер является самостоятельной сущностью и не зависит от применяемого радиоприёмного оборудования. Это значит, что SDR-приёмник сервера надо предварительно настроить как одну независимую сущность, а сервер – как другую. Такая конфигурация накладывает определённые ограничения на функционал. Вы, как клиент, подключающийся к серверу не имеете возможности настроиться на любую интересную вам частоту, а только туда, куда настроен сервер и только в той полосе, которое позволяет полоса оцифровки. Так, если на сервере стоит приёмника типа Softrock со звуковой картой полосой 96кГц или 192кГц клиент сможет настраиваться только в данной полосе. Гораздо интереснее в этом отношении приёмники SDRPlay и RTL-SDR с полосой оцифровки до 2МГц.

Все настройки сервера описываются несколькими командами в файле конфигурации:

- 1) Глобальные настройки доступа к вашему серверу
- 2) Описание расположения файлов
- 3) Команды настройки Java-ядра
- 4) Настройки появления вашего сайта на агрегаторе приёмников websdr.org
- 5) Настройки конфигурации отдельно по диапазонам.

Отдельно настраивается файл с метками на шкале частот.

Более подробно все команды описаны в приложении-переводе документации.

Следует отметить, что часть команд можно безболезненно пропустить. И, если вы решили запустить сервер websdr локально во внутренней сети или для друзей, то обязательно ставьте команду запрещения публикации вашего сервера на сайте websdr.org, а также помещайте адрес в заглушку localhost !

Существуют некоторые особенности работы сервера и настройки ОС.

По всей видимости, по причине того, что автор сервера перестал поддерживать своё детище, существуют несколько неочевидных особенностей о которые сегодня можно долго биться, не обладая глубокими знаниями в Linux.

- 1) Крайняя бесппроблемная версия ОС Ubuntu, на которой стоит ставить сервер – версия 16.04 (Xenial). На 17 и 18 версиях есть проблемы с библиотекой libpng12, и что бы её корректно установить, нужно специально править файл ресурса и тянуть библиотеку с ресурса 16 версии. Более подробно будет описано отдельно. На 19 версии Ubuntu запустить сервер не удалось. Углублённое гугление вопроса показало, что в новом релизе какие-то проблемы с векторами линковки файлов. Возможно, это как-то решается...
- 2) Практически все файлы для работы сервера уже в ОС присутствуют, понадобится только установить библиотеки libfftw3 и libasound.
- 3) Для Raspberry Pi крайняя бесппроблемная версия Jessie. Запускать ОС на 3-ей Малине или Малине-Zero крайне желательно минимальную консольную версию, на которой больше всего остаётся свободных вычислительных ресурсов. На Stretch версии Расбиана, возможно, понадобится отдельно установить файл libcrypto.so.1.0.0 Buster-версию Расбиана не стал пробовать.
- 4) Если сервер строится на базе RTL-SDR свистка на KB диапазон в режиме DDC, то драйвера необходимо собирать из исходников. (Описано будет ниже)
- 5) Если на полноценном Raspberry PI сервера может без проблем работать по LAN, то для Raspberry Zero сервер работает только через WiFi. Настройка беспроводной сети выходит за рамки этой статьи и может вызвать у некоторых проблемы, потому, за эксперименты с нулевой малинной стоит браться только в том случае, если вы достаточно глубоко понимаете в Linux, ну или гугел вам в помощь.
- 6) Сервер умеет работать внутри вашей локальной сети. Для запуска сервера в большой мир, вам необходимо иметь белый IP-адрес и, желательно хостинг. Впрочем, на первом этапе, или, если вы планируете делать свой websdr для ограниченного круга лиц, с хостингом можно не заморачиваться. Существуют сервисы туннелирования потоков типа NoIP или платные VPN-сервисы.
- 7) При небольшом количестве народа (опробовано до 20 человек) сервер нормально работает через обычный Wi-Fi.
- 8) Возможна работа нескольких независимых серверов под одним IP адресом, но на разных портах.
- 9) Ну, и, надо понимать, что данное описание не является полноценным исчерпывающим руководством. Автор этого текста даёт только направление для самостоятельных опытов и экспериментов. Если в процессе запуска сервера у вас появятся трудности, вопросы или ошибки, то на поиск решения проблем придётся самостоятельно потратить время в гугле или поискать знакомых гуру по Linux, желательно в РЛ сообществе.

В приложении будут представлены образцы вариантов запуска сервера на RaspberryPi и обычном компьютере на основе RTL-SDR приёмников. Все манипуляции с папками и файлами на Flash-накопителе для Raspberry удобно осуществлять на отдельном полноценном Linux-компьютере. Если вы в Linux только начинающий, то можно с начать опробование-обучение на виртуальной машине, например VBOX или VMWARE.

ПРИЛОЖЕНИЕ 1: Перевод документации из состава сервера.

Вводная инструкции по установке сервера WebSDR

Данная инструкция описана и проверена на версиях Ubuntu 16.04 и 18.04

1. Желательно создать новую учетную запись пользователя для WebSDR. В этом нет технической необходимости, но предпочтительно с точки зрения безопасности. Это ограничивает ущерб в случае, если программное обеспечение сервера WebSDR будет иметь ошибку безопасности.
2. Убедитесь, что вновь созданная учетная запись WebSDR имеет разрешение на доступ к звуковой карте; Обычно это делается путем добавления пользователя в Аудио группу:

```
adduser websdr audio
```

3. В домашнем каталоге этой учетной записи распакуйте файл websdr-dist10a.tgz. В дальнейшем я буду предполагать, что этот домашний каталог: `/home/websdr`

Распаковка этого файла создает полную структуру каталогов, необходимую для работы WebSDR:

`/home/websdr/pub` - публичные файлы (специфичные для сайта).

`/home/websdr/cfg` - содержит файлы конфигурации для конкретного сайта.

`/home/websdr/dist11` - файлы, не относящиеся к сайту, для этой версии.

`/home/websdr/dist11/doc` - содержит документацию.

`/home/websdr/dist11/log` - содержит подробные файлы журнала.

`/home/websdr/dist11/pub2` - другой набор общедоступных файлов, не зависящий от сайта.

`/home/websdr/dist11/pub2/tmp` - временные публичные файлы.

4. Скопируйте файл `/home/websdr/dist11/websdr.cfg` в `/home/websdr/cfg` а затем используйте ваш любой текстовый редактор (например Nano) для редактирования файла websdr.cfg

Помните! Копия в cfg, не есть оригинал в dist11.

Документация для этого файла описана в самом файле, в виде строк комментариев (строки начинающиеся с #).

5. Скопируйте файл `/home/websdr/dist11/pub2/index.html` в `/home/websdr/pub` и снова используйте текстовый редактор для редактирования файла index.html

Помните! Копия pub не есть оригинал в pub2.

Оригинальный файл отображает страницу по умолчанию в классическом виде, но вы можете настроить текстовую информацию, которую страница содержит на свой вкус;

два соответствующих места в файле помечены как TODO, так что вы можете быстро найти их с помощью текстового редактора.

Позже, когда после настроек страница заработает нормально, вы можете настроить этот файл так чтобы страница выглядела красочно и привлекательно, вы можете добавить больше своей информации на вашу страницу и графику.

6. Если вы хотите поддерживать мобильные устройства, скопируйте файл `/home/websdr/dist11/pub2/m.html` в `/home/websdr/pub` и используйте текстовый редактор для редактирования, если это необходимо.

Внимание! Не помещайте на мобильной странице много картинок: эту страницу будут просматривать на маленьких экранах телефонов!

7. Для запуска сервера WebSDR, перейдите в папку `/home/websdr/dist11` и наберите `./websdr`

(Или `./websdr64`, если вы используете 64-битную систему Linux, или `./websdr-gpi` если ваш сервер запущен на Raspberry Pi).

Если прежде все было правильно настроено в звуковом устройстве или правильно сконфигурирован RTL-TCP сервер, или звуковое устройство Afedri, то сервер должен заработать!

Важные примечания:

- Если вы хотите запустить сервер на порту 80, чтобы номер порта мог быть исключенным из URL (то есть вы получаете <http://websdr.example.org/> вместо чего-то вроде <http://websdr.example.org:8901/>), вам нужно сделать бинарный сервер setuid корневым.

Это можно сделать, перейдя в `/home/websdr/dist11` и набрав:

```
sudo chown root websdr
```

```
sudo chmod 4510 websdr
```

- Чтобы свести к минимуму риски безопасности, сервер WebSDR написан так, что он использует только привилегии root. Например, открыть сокет TCP или сбросить его в любое время.
- Серверу нужно несколько сторонних библиотек. Большинство из них будут установлены по умолчанию в вашем дистрибутиве Linux, но двух из них может не быть, а именно:

`libfftw3f` (то есть, версия 3 библиотеки FFTW, скомпилированная для float), а также

`libpng12` (т.е. версия 1.2 библиотеки PNG).

Обычно эти библиотеки доступны через ваш дистрибутив и систему управления. В случае Debian и Ubuntu вы можете установить эти пакеты отдельно, используя следующую команду (как root):

```
Sudo apt-get install libfftw3-3
```

ВНИМАНИЕ! Конкретно с библиотекой `libpng12` есть проблемы и как её поставить будет описано отдельно.

- Смотрите файлы в `/home/websdr/dist11/doc/` для получения дополнительной информации; в частности, `config.txt` объясняет формат файла конфигурации.

Настройка файла конфигурации websdr.cfg

Основной функционал сервера WebSDR настраивается через один файл с именем 'Websdr.cfg'. Этот файл содержит первые глобальные настройки, а затем настройки для каждого диапазона. Каждая строка начинается с ключевого слова, за которым следует один или несколько аргументов, каждый подробно описан ниже.

Объявление данных для глобальной конфигурации:

Описание доступа к серверу

- **myhost x**

Привязывать сервер только к этому адресу; полезно, если ваш сервер имеет несколько IP адресов.

Примечание: если (x) является IPv4-адресом, пропишите ему в начале ::ffff:, чтобы он выглядел как адрес IPv6.

- **tcpport x**

Установите номер порта TCP, на котором должен работать сервер.

(Это, например, 8901 на моём адресе <http://websdr.ewi.utwente.nl:8901/>)

По умолчанию установлен 80 порт, он также используется веб-браузерами по умолчанию, когда ни один из них не используется.

Примечание: для этого требуется запустить websdr от имени root - пользователя.

- **chroot x**

После запуска выполните команду chroot () для x.

Это может быть использовано в качестве меры безопасности (хотя я не делаю это на моем собственном сервере). Это требует, чтобы websdr был setuid root, но те привилегии отбрасываются сразу же, как только возможно.

- **rawpassword x**

Установите пароль, чтобы ограничить доступ к потоку необработанных данных. По умолчанию пароль отсутствует. (Это не угроза безопасности.)

- **donttrustlocalhost**

Включение этой строки отключает доступ администратора даже с текущей машины. Это нужно только в том случае, если ваш websdr работает на компьютере, который делится с пользователями, которым вы не доверяете.

Упомянутый «доступ администратора» включает в себя две вещи:

- используется интерфейс администратора в sysop.html;

- разрешает доступ к приемнику, даже если был достигнут предел максимального количества подключённых пользователей.

- **dotrustlocalnet**

Включение этой строки включает доступ администратора со всех компьютеров в диапазоне локальных IP (10. *. *. * и 192.168. *. *).

Используйте это, только если вы доверяете всем пользователям в вашей локальной сети.

- **dotrust x**

Разрешить доступ администратора с адреса x (вы можете включить здесь несколько адресов разделенных пробелами).

Описание расположения файлов

- **chatboxlogfile x**

Лог всех сообщений чата в файле с именем x.

- **logfileinterval x**

Установка времени (в секундах), после которого запускается следующий файл журнала, а предыдущий архивируется. По умолчанию стоит значение 3600 (т.е. один час).

Если установить значение 0, файл журнала не будет записан, но сообщения журнала будут отправлены в поток stderr (системный поток вывода журнала ошибок).

- **logdir x**

Установка имени каталога, содержащего файлы журнала, значение x.

По умолчанию это папка log

- **public x**

Установка имени каталог, содержащий публичные файлы (html и т. Д.), Равным x.

По умолчанию это папка ../pub

- **public2 x**

Установка имени второго каталог, содержащего публичные файлы (html и т. Д.), Равным x.

В этом каталоге также выполняется поиск включений на стороне сервера.

По умолчанию это папка pub2

Настройки FFT обработчика

- **maxusers x**

Ограничивает количество одновременно подключённых пользователей до x.

(Пользователи из localhost или диапазона локальной сети все еще могут быть разрешены сверх этого предела; смотрите настройки dontrustlocalhost и dotrustlocalnet)

- **slowdownusers x**

Замедление водопада допускаются только тогда, когда количество слушателей превышает значение в настройках **maxuser**.

Полезно для экономии ресурсов процессора на медленных компьютерах или пропускной способности сети, а также, на медленных соединениях.

- **slowdownusers2 x**

Замедление водопада до половины нормальной минимальной скорости, когда число слушателей превышает x.

- **fftplaneffort x**

Установка для планирования производительности библиотеки FFTW, где x может быть:

- 0 для FFTW_ESTIMATE
- 1 для FFTW_MEASURE
- 2 для FFTW_PATIENT
- 3 для FFTW_EXHAUSTIVE

Большая цифра замедляет запуск программы, но может привести к снижению нагрузки на процессор, потому что библиотека пытается оптимизировать БПФ расчет. По умолчанию стоит 0.

- **waterfallformat x**

Установка размера данных для потока данных водопада.

x = 0 дает старый несжатый формат 8 бит на пиксель.

x = 8 использует в среднем 2,9 бит на пиксель.

x = 9 теперь используется по умолчанию: сжатие между 1,5 и 2,3 битами на пиксель с очень небольшим количеством артефактов.

- **audioformat x**

Установка величины сжатия для аудиоданных. X может быть от 0 до 3.

Чем выше x, тем больше будет шум квантования. Обычно x = 0 (это значение по умолчанию) должно быть хорошим; но если пропускная способность вашей сети ограничена, вы можете попробовать другие значения.

(Я использовал x = 2 в моей домашней системе без каких-либо жалоб пользователей; x = 3 было признано слишком шумным.)

- **initial x y**

Установка начальной частоты x кГц в модуляции y (usb/lb/am/cw/fm).

- **allowwide**

Разрешение расширить полосы пропускания сигналов SSB и AM за пределы стандартных значений 4 и 8 кГц; полезно для лучшего качества звука на AM вещательных сигналах, но увеличивает нагрузку на интернет-связь. Кроме того, это правильно поддерживается только для аудио на основе HTML5; (при использовании Java наблюдается алиасинг продуктов).

Примечание: изменение полосы пропускания во время записи также не работает (файлы WAV не поддерживают изменение частоты дискретизации).

- **idletimeout x**

Установка времени ожидания в x секунд для не активных пользователей (то есть пользователи, которые не изменяют частоту настройки или не предпринимают никаких других действий на x секунд будут отключены).

Может быть полезно, если ваш сайт популярен и имеет ограниченную пропускную способность Интернета.

Настройки, связанные с появлением вашего персонального сервера на странице <http://www.websdr.org>

- **noorgserver**

Включение этой строки предотвращает включение вашего сервера в список на сайте <http://www.websdr.org/>

- **hostname x**

Установите свое собственное имя хоста. Если установлено, это имя будет отображаться на сайте <http://www.websdr.org> вместо вашего IP-адреса.

- **org**

Установка ваших данных для отправки на сервер websdr.org.

org qth x - установка вашего QTH-локатора

org description x - задайте краткое описание вашего сайта

org logo x - установить имя файла логотипа

org email x - ваш адрес электронной почты

Примечание: ваш адрес электронной почты НЕ ПУБЛИКУЕТСЯ на сайте websdr.org

Он нужен автору на случай с кем связаться, если вдруг на сервере есть проблема, и кому отправлять обновления программного обеспечения.

Из вышесказанного, логотип не является обязательным; остальные данные необходимы.

Объявление конфигурации для каждого диапазона:

- **band x**

Старт конфигурации для диапазона X.

Должно быть первое объявление настроек для диапазона.

- **device x**

Устанавливает устройство ввода в x, (например /dev/dsp0)

Имена устройств ALSA также могут использоваться с префиксом «\$», например,

\$ hw: 0,0,0

device x y

Альтернативный синтаксис для использования со звуковой картой с несколькими входами.

y - имя устройства, например /dev/dsp0

x - однозначное число: 0 для первых двух входов, 1 для следующих двух входов и т. д.

Все диапазоны, использующие звуковую карту, должны иметь одинаковую частоту дискретизации, и они не может иметь настройки «poniq» (см. ниже) и устройств ALSA.

- **device !rtlsdr x:y z**

Устанавливает устройство ввода в RTL-SDR (USB-ключ DVB-T), через драйвер rtl_tcp, работающий на хосте x, на порту y, с коррекцией ppm z. (Z совместно с настройками rtl_tcp)

Поддерживаемые частоты дискретизации для использования с RTL-SDR:

256000, 512000, 768000, 1024000, 1536000, 2048000 и 2880000 Гц.

Для использования RTL-SDR свистка с повышающими/понижающими преобразователями, см. настройки «**Progfreq**» в другом месте этого документа.

Примечание: у этих свистков довольно маленький динамический диапазон. Как следствие, когда в полосе пропускания приёмника появятся сильные сигналы, то все принимаемые в это время сигналы исчезнут в шумовой полке. Для WebSDR это не есть хорошо, потому что идея WebSDR заключается в том, что каждый пользователь может настраиваться независимо на любой сигнал в пределах полосы оцифровки. Поэтому, пожалуйста, запускайте WebSDR с RTL-SDR свистка только после того, как вы убедились, что в вашем QTH свисток не перегружается. (Решается установкой оптимального усиления в настройках при старте rtl_tcp)

- **extrazoom x**

Максимальная глубина увеличения на x шагов. Каждый шаг удваивает максимальное разрешение, но уменьшает вдвое максимальное (и по умолчанию) скорость прокрутки водопада. По умолчанию 0. Большее значение может быть полезно, чтобы обеспечить более длительную интеграцию для длительной несущей, но WebSDR не очень хороший инструмент для этого ...

- **samplerate x**

Установите частоту дискретизации (в Гц).

- **centerfreq x**

Установка центральной частоты (в кГц)

Обратите внимание, что программное обеспечение WebSDR не умеет управлять иным SDR-оборудованием (кроме RTL-SDR). В случае применения SDR с фиксированной частотой (Например приёмник SoftRock), вы просто описываете центральную частоту, определяемую кварцевым кристаллом SDR.

Если у вас есть SDR с синтезатором, вам нужно некоторое внешнее программное обеспечение, чтобы установить частоту аппаратного обеспечения SDR на той же частоте, какой вы опишите здесь.

- **progfreq x**

В случае, если у вас есть внешнее оборудование для SDR (например, UP/DN –конвертер для RTL-SDR свистка) программное обеспечение которого WebSDR контролирует частоту (т.е. тот же RTL-SDR свисток), эта настройка может использоваться для указания, на какую частоту (в кГц) отличным от значения **centerfreq** должно быть подключено оборудование или быть программным. Это полезно в случае использовать с повышающим / понижающим преобразователем.

- **noniq**

Входные данные обрабатываются как сигнал основной полосы (моно), а не как I / Q квадратурный (стерео) сигнал. Это полезно для приемника VLF, с антенной подключённой напрямую к звуковой карте.

При установленном noniq значение centerfreq интерпретируется не как центр полосы пропускания, но как левый конец полосы пропускания (т.е. 0 в случае VLF).

- **swapiq**

Замена местами каналы I и Q (левый и правый).

Используйте это, если полоса пропускания окажется зеркальной.

- **balance x**

Считать данные балансировки из файла x.

Используйте это для компенсации незначительных различий в каналах I и Q в результате чего сигналы появляются «зеркально».

См. Отдельную документацию в doc /iq-balance.txt.

- **antenna x**

Краткое описание антенны, которая будет указана на сайте <http://www.websdr.org>.

- **delay x**

Искусственно задерживать сигналы на x секунд.

Полезно, чтобы предотвратить мошенничество в соревнованиях, если ваш приемник такой особенный, что его использование даст участникам несправедливое преимущество (например, очень большое блюдо для EME).

- **gain x**

Введите x дБ усиления (может быть отрицательным).

Это не делает слышимым слабый сигнал; это только сдвигает на водопаде цветовую шкалу и S-метр, так что вы можете использовать это для калибровки S метр. (На RTL-SDR свистке оптимальное значение -30...-50)

- **equalize z**

Считать уравнивающие данные из файла x.

Это в основном частотно-зависимая версия установок вышеуказанного «усиления». Может использоваться для компенсации незначительных изменений усиления каналы звуковой карты.

Файл имеет очень простой формат: каждая строка имеет частоту в кГц и соответствующая коррекция усиления в дБ.

Программное обеспечение будет интерполировать между заданными частотами. Частоты "аудио" / "основной полосы частот", а не радио частоты.

- **hpf x**

Установите частоту среза входного фильтра верхних частот равной x Гц.

Этот фильтр отключает смещение постоянного тока и, возможно, другие низкочастотные шум (например, сетевой шум) на входе звуковой карты.

Это всего лишь фильтр первого порядка, поэтому он не очень крутой. Установка его в 0 полностью удаляет фильтр. По умолчанию 500 Гц.

- **noiseblanker x**

Включает простой шумоподаватель, где x устанавливает пороговый коэффициент.

Чем выше x, тем выше порог, выше которого всплеск шума должен подняться, чтобы быть заглушенным; значение порядка 80 имеет смысл, но эксперименты могут быть необходимы.

Из-за своей простоты этот шумоглушитель, вероятно, только полезен если у вас достаточно сильные местные источники шума, например, «электрический забор».

Информация о станциях для меток на частотной шкале

Если вы хотите поместить некоторые метки на шкале частот, например, чтобы пометить маяки или другие постоянные станции, представляющие особый интерес, вы можете поставить их в файле с именем 'stationinfo.txt' в каталоге cfg/

Формат этого файла очень прост: каждая строка имеет частоту в кГц, после чего следует краткое описание (например, позывной).

Вы также можете включить режим (am/usb/lb/cw/fm) сразу после частоты.

Например: 14100cw Международные маяки

Этот файл перечитывается каждую минуту, поэтому, если вы вносите изменения, вы делаете не нужно перезагружать сервер.

Исправление дисбаланса I/Q потока

Аналоговые SDR, как правило, не имеют идеального баланса в I/Q каналах: два канала (называемые I и Q, или левый и правый вход звуковой карты) не могут иметь точно такое же усиление или сдвиг фазы.

Это приводит к зеркальному отображению сигналов.

Они могут быть в некоторой степени подавлены в программном обеспечении. В настоящее время это процедура не автоматическая в WebSDR, а только полуавтоматическая.

- 1) Подключите генератор сигналов к антенному входу SDR; установить генератор для производства немодулированной несущей в SDR полосе пропускания и установите ее уровень так, чтобы сигнал был очень сильным (появляются зеркальные сигналы), но не настолько сильно, чтобы системные клипы. (Отсечение можно рассматривать как появление даже больше сигналов на дисплее водопада.)

Установите частоту близко к нижней границе полосы пропускания, но все еще внутри диапазона.

- 2) Используйте веб-браузер или программу 'wget', чтобы получить следующее

URL: `http://your.websdr.hostname.here/~~iqbalance0`

где последняя цифра должна быть 0 для первой полосы, 1 для второго диапазона и т. д.

- 3) Пока поступают данные, медленно увеличивайте уровень генератора частота, пока вы не будете чуть ниже верхнего предела полосы.
- 4) Остановить получение данных.

Сохраните файл в каталоге `websdr/cfg` под подходящим именем. (например, `/home/websdr/cfg/iq-80m.txt`).

При необходимости очистите данные (см. Ниже).

- 5) В вашем `websdr.cfg` добавьте строку в соответствующий раздел диапазона, например,:
`balance filename.dat`

где `filename.dat`, очевидно, является именем, под которым вы сохранили данные.

- 6) Перезапустите `websdr`, например, через `sysop.html`.

Этот странный URL называется `~~iqbalance` производит через равные промежутки времени один строка текста, содержащая данные коррекции I/Q для частоты, которая в этот момент содержит самый сильный сигнал (который должен быть сигнал от вашего генератора сигналов).

Первый столбец представляет частоту, следующие два столбца поправочные коэффициенты, следующий за амплитудой тестового сигнала, а последний вспомогательная величина, которая более не актуально. В работе используются только первые три столбца.

Файл должен быть в порядке увеличения частоты; если нужно, отсортируйте его (например, используя ``sort -n``).

Обратите внимание, что вы не включаете какие-либо точки данных, записанные при генераторе сигналов был только за пределами полосы: из-за несовершенного фильтры сглаживания в звуковой карте, такие сигналы могут быть получены, но дал бы очень неправильные I/Q исправления.

WebSDR на Raspberry Pi

Начиная с выпуска dist10, программное обеспечение WebSDR поставляется с версией JAVA-машины для Raspberry Pi.

Тем не менее, эта версия (или, скорее, Raspberry Pi) имеет некоторые ограничения, которые описаны ниже. Пока нет гарантии что сервер WebSDR будет работать на Raspberry Pi правильно.

Монтаж

Для установки программного обеспечения требуется Raspberry Pi с запущенным образом Rasbian, но в остальном происходит точно так же, как на обычном ПК с Linux, за исключением того, что в конце вы стартуете не `./websdr`, а `./websdr-rpi`.

Кроме того, будьте осторожны, чтобы выбрать правильное устройство ввода; по умолчанию, например, `$ hw: 0,0,0` не будет работать, потому что это относится к встроенному в Raspberry Pi звуковому устройству, которое не имеет входного аудиоканала.

Если программное обеспечение websdr жалуется на то, что IPv6 не поддерживается, загрузите модуль ядра IPv6, набрав:

```
sudo modprobe ipv6
```

Вы можете сделать это во время загрузки, добавив `ipv6` в список в `/etc/modules`.

Поддерживаемые SDR устройства

Поскольку Raspberry Pi не имеет аудиовхода, вы не можете просто подключить SDR в стиле softrock. Вместо этого нужно будет что-то подключить через USB.

У вас есть два варианта:

- использовать звуковую карту USB со стерео-входом и подключить приёмник в стиле SDR-softrock.
- использовать SDR, который имеет USB-соединение и представляет из себя компьютер как стандартное звуковое устройство (не все подключенные к USB SDR!). FunCubeDongle ?
- 8-битный RTL-SDR приёмник

Скорость и возможные проблемы

Как отмечалось выше, Raspberry Pi не очень быстрое устройство.

При использовании FunCubeDongle Pro с частотой дискретизации 96 кГц загрузка процессора составляет чуть более 50%.

Это был не разогнанный R-Pi; теперь есть способы слегка разогнать его без гарантии, так что частота дискретизации 192 кГц будет работать.

Загрузка процессора не единственная проблема.

Кажется, есть проблемы с синхронизацией с шиной USB, которые могут вызвать пропуск пакетов в аудиопотоке. Кроме того, кажется, доступ к SD карта блокирует шину USB, вызывая сброс или задержку данных.

Некоторые возможности уменьшить эти проблемы:

- Уменьшите скорость USB, установив `dwc_otg.speed = 1` в `/boot/cmdline.txt` и перезагрузите. Из того, что я знаю, это, с одной стороны, улучшает USB время, но может также привести к тому, что клавиатура и мышь перестанут не работать правильно.
- Уменьшите запись файлов журнала с помощью WebSDR, установив `logfileinterval 0` в `websdr.cfg`.
- Используйте USB-флешку для каталога `websdr` вместо SD-карты.

ПРИЛОЖЕНИЕ 2: Пример по запуску сервера на обычном компьютере.

Перед установкой и запуском сервера на компьютере должна быть установлена 32 или 64 битная операционная система Linux. Желательно x64. Это может быть любая Дебиано-совместимая ОС. Автор опробовал на своём компьютере обычные версии Ubuntu 16 x64 и 18 x64 LTS и версии x64 Mini. Сервер работает так же и на ОС Debian. Предварительно должен быть настроен выход в интернет. Если на вашем Роутере адреса раздаются по DHCP, то потом надо будет настроить проброс IP-адреса и порта в большой интернет. При небольшом количестве народа возможна работа сервера через W-Fi, но, лучше делать нормальное проводное соединение.

Перед установкой сервера для обеспечения большего удобства желательно (но не обязательно) установить некоторый программный инструментарий. Я использую:

- FAR2Linux (Качается с интернета deb-пакет)
- MC : `sudo apt-get install mc`
- HTOP : `sudo apt-get install htop`
- SSH-сервер : `sudo apt-get install ssh`

FAR и MC – нортоноподобная среда работы с файлами

HTOP – показывает состояние системы, памяти и процессора, запущенные процессы.

По SSH-удобно работать с сервером удалённо.

- 1) Устанавливаем драйвера для RTL-SDR приёмника.

```
sudo apt-get update
sudo apt-get install rtl-sdr
```

Если необходимо приёмник запустить в режиме Direct-Samples, то драйвера надо собрать будет самостоятельно. Качаем с гитхаба крайнюю версию исходников и собираем их.

```
sudo apt-get install libusb-1.0-0-dev git cmake
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr/
mkdir build
cd build
cmake ../ -DINSTALL_UDEV_RULES=ON
make
sudo make install
sudo cp ../rtl-sdr.rules /etc/udev/rules.d/
sudo ldconfig
```

- 2) Заблокируем переопределение RTL драйверов по умолчанию

```
cat <<EOF >no-rtl.conf
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
EOF
sudo mv no-rtl.conf /etc/modprobe.d/
```

- 3) Перезагрузим компьютер.

```
sudo reboot
```

- 4) Теперь можно вставить RTL-SDR приёмник (или несколько) в компьютер и протестировать их.

```
rtl_test
```

Если с драйверы встали корректно, то мы увидим следующий текст о том, что тест пройден успешно:

```
user@websdr:~$ rtl_test
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Detached kernel driver
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7
20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0 4
9.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
```

- 5) Установим необходимые для работы сервера библиотеки

```
sudo apt-get update && sudo apt-get install libx11-
xcb1 libxss1 libasound2 libxkbfile1
sudo apt-get install libfftw3-dev libfftw3-3 libpng-dev
sudo apt-get install libasound2 libasound2-dev
sudo apt-get install -f libpng-dev
```

Примечание: Выше установленные библиотеки работают корректно на Ubuntu 16 и на этом все заморочки с окружением заканчиваются. Если вы запускаете сервер на Ubuntu 18 то нас ждут дополнительные танцы с бубнами!

Ядро сервера websdr написано довольно давно и сегодня работает не совсем корректно. При запуске оно КАТЕГОРИЧЕСКИ требует libpng12. Сегодня, библиотека имеет совсем другую обновленную версию (кажется 1.6), мир использует libpng-dev. И то, что на дворе конец 2019 года никого не волнует. Попытка обойти дурацкое требование методом линковки на версию посвежее - ничего не дала, после того как приложение находит библиотеку, оно проверяет версию. И если версия не 12, то сервер начинает истерить и материться. Так что, только libpng12, только хардкор!

Для установки старой библиотеки надо подключить источник репозитория из 16 версии Ubuntu:

```
sudo nano /etc/apt/sources.list
```

Добавляем в источники ещё один адрес и сохраняемся:

```
#### Manually Added sources
## source for libpng12-0 package
deb http://mirrors.kernel.org/ubuntu/ xenial main
```

Делаем апдейт системы и устанавливаем библиотеку:

```
sudo apt-get update
sudo apt-get install libpng12-0
```

После этого ещё раз правим источник, убираем новый ресурс или закомментируем его.

Собственно, на этом окружение можно считать настроенным. Для верности можно перезагрузить ещё раз компьютер.

```
sudo reboot
```

- 6) Далее, необходимо скопировать папку с сервером websdr в любой каталог. Удобно в /home или /user.
- 7) Правим файл websdr.cfg для настройки вашего сервера. Ниже приведён пример простейшей конфигурации для 2х приёмников.

```
maxusers 80
fftplaneffort 2
allowwide
tcpport 8901
noorgserver
initial 145000 fm

#BAND 2m
band 2m
device !rtlsdr 192.168.1.200:1001
samplerate 2048000
centerfreq 145000
antenna GP 3x5/8
gain -25
extrazoom 1

#BAND 70cm
band 70cm
device !rtlsdr 192.168.1.200:1002
samplerate 2048000
centerfreq 433500
antenna GP 5x5/8
gain -25
extrazoom 1
```

Сохраняемся и выходим.

- 8) Переходим в папку /websdr/dist и запускаем сервер RTL-SDR приёмника.

```
user@websdr:~$ cd /home/user/websdr/dist
```

Запуск возможен в 3х вариантах представления команд.

А) Обычный запуск RTL-TCP сервера:

```
rtl_tcp -a 192.168.1.200 -p1001 -f145000K -g0 -s 2048000 -d0 &
Нажимаем клавишу «Enter» и задаём команду для для второго приёмника
rtl_tcp -a 192.168.1.200 -p1002 -f433500K -g0 -s 2048000 -d1 &
Нажимаем клавишу «Enter» для ввода следующей команды
```

Б) Запуск RTL-TCP сервера с расширенным набором свойств:

```
rtl_tcp -b 3 -n 1000 -a 192.168.1.200 -p1001 -f145000K -g0 -s 2048000 -d0 &
Нажимаем клавишу «Enter» и задаём команду для для второго приёмника
rtl_tcp -b 3 -n 1000 -a 192.168.1.200 -p1002 -f433500K -g0 -s 2048000 -d1 &
Нажимаем клавишу «Enter» для ввода следующей команды
```

В) Запуск RTL-TCP сервера укороченным методом, при котором параметры приёмника подхватываются автоматически из файла конфигурации websdr.cfg

```
rtl_tcp -a 192.168.1.200 -p1001 -d0 &  
Нажимаем клавишу «Enter» и задаём команду для для второго приёмника  
rtl_tcp -a 192.168.1.200 -p1002 -d1 &  
Нажимаем клавишу «Enter» для ввода следующей команды
```

Любой из вариантов запуска является правильным и зависит от ваших предпочтений и понимания оптимизации нагрузки.

Скрин успешного запуска на примере 1 приёмника:

```
user@websdr:/home/websdr/dist$ rtl_tcp -b 3 -n 1000 -a 192.168.1.200 -p1001 -  
f145000K -g0 -s 2048000 &  
  
user@websdr:~$ Found 1 device(s):  
 0: Realtek, RTL2838UHIDIR, SN: 00000001  
  
Using device 0: Generic RTL2832U OEM  
Detached kernel driver  
Found Rafael Micro R820T tuner  
[R82XX] PLL not locked!  
Tuned to 145000000 Hz.  
rtl_tcp bind error: Permission deniedlistening...  
Use the device argument 'rtl_tcp=192.168.1.210:1001' in OsmoSDR (gr-  
osmosdr) source  
to receive samples in GRC and control rtl_tcp parameters (frequency, gain, ...).  
  
user@websdr:~$
```

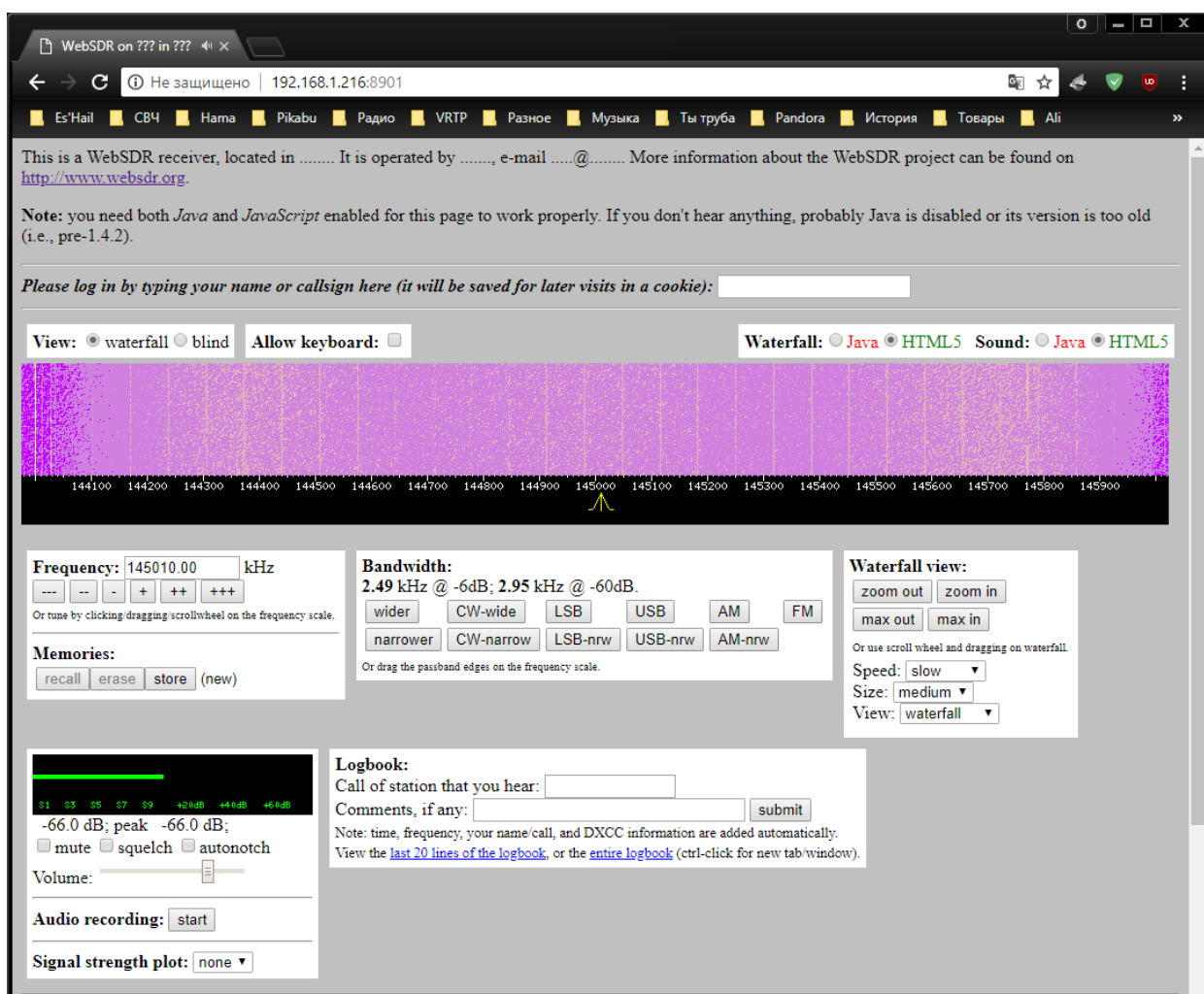
9) Из папки /websdr/dist запускаем сам websdr-сервер от root

```
sudo ./websdr64
```

В случае успешного запуска увидим следующее:

```
user@websdr:/home/websdr/dist$ sudo ./websdr64  
  
>>> WebSDR server (20140718.1716-64)<<<  
  
Copyright 2007-2014, Pieter-Tjerk de Boer, pa3fwm@websdr.org  
  
Planning FFTs for band 0... done.  
Writing frequency axis images... done  
  
Initialization completed; server is now running.
```


10) Запускаем на любом компьютере браузер Firefox или Chrome, вводим в строку поиска IP-адрес и порт вашего сервера: 192.168.1.200:8901



Дальнейшая работа с сервером сводится к настройкам HTML разметки, стартовых условий по частоте и модуляции, реализация собственных фантазий, связанных с оформлением сайта. Достойные примеры приведены в начале этой статьи.

Примечание 1: Иногда требуется настроить полный доступ к файлам и папкам сервера:

```
Chmod 777 -R /home/user/websdr/
```

Примечание 2: При первом запуске сервера происходит калибровка FFT-ядра. В это время на экране будет виден просчёт пакетов в виде бегущих строк. Это нормально. При последующих запусках, так же происходит просчёт, но быстро. У автора, перед успешным запуском, сервер считает где-то до 70 единиц. Если вдруг сервер начинает считать до бесконечности и/или зависать и/или аварийно завершаться – это значит, что неправильно настроен файл конфигурации или, если у вас старый компьютер, то он работает на пределе возможностей.

Помните, что большинство ошибок при запуске сервера имеют характер простейших синтаксических ошибок и банальной невнимательности!

ПРИЛОЖЕНИЕ 3: Пример по запуску сервера на Raspberry Pi Zero

Настройка websdr-сервера на Raspberry Pi Zero приводится кратко, т.к. она мало отличается от настроек на обычном компьютере, подразумевая, что вы уже ознакомились с Приложением 2.

Для примера, здесь показано, что возможен запуск сервера в самом минимальном варианте на Raspberry Pi Zero. Это значит, что на обычной Raspberry Pi 3 сервер websdr будет работать 100%. Ограничения в работе websdr-сервера на Raspberry Pi Zero в минимально возможной рабочей полосе – 256кГц и для ограниченного круга клиентов – максимум, 3...5 человек.

Для Raspberry Pi крайняя беспроблемная версия ОС - Jessie. Запускать ОС на 3-ей Малине или Малине-Зеро крайне желательно минимальную консольную версию, на которой больше всего остаётся свободных вычислительных ресурсов. На Stretch версии Расбиана, возможно, понадобится отдельно установить файл libcrypto.so.1.0.0 (Файл будет лежать в архиве в отдельной папке) Buster-версию Расбиана не стал пробовать, по причине основы её на крайней версии Debian, с которой websdr отказывается работать.

Пример с Raspberry Pi Zero немного более сложен, чем Raspberry Pi 3, т.к. на плате всего 1 micro-USB-порт, нет LAN-разъёма, но есть WiFi. Потому, нам понадобится сделать некоторые танцы с бубном, прежде чем мы начнём работу с сервером websdr. Множество инструкций вы с лёгкостью найдёте в гуглах, но автор всё же покажет, как это делал у себя. Для этого, на первом этапе понадобится подключить через переходник USB –клавиатуру и монитор. Все операции придётся делать своими ручками, поэтому, запаситесь терпением. ☺

На свежи установленной системе надо запустить WiFi и подключить к вашей локальной Wi-Fi сети. Сделать это можно легко, прописав настройки в файл wpa_supplicant.conf

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Наполнение файла должно быть следующим:

```
country=RU
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="You Wi-Fi Name"
    psk="You Wi-Fi password"
}
```

Т.е. в строках SSID и PSK вам надо прописать имя и пароль для вашей локальной сети.

Перезагружаемся:

```
sudo reboot
```

После этого проверяем, есть ли у нас подключение к Wi-Fi

```
ifconfig
```

Вы должны увидеть ваше соединение:

```
root@raspberrypi:/# ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:2813 errors:0 dropped:0 overruns:0 frame:0
            TX packets:2813 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1
            RX bytes:74945049 (71.4 MiB)  TX bytes:74945049 (71.4 MiB)

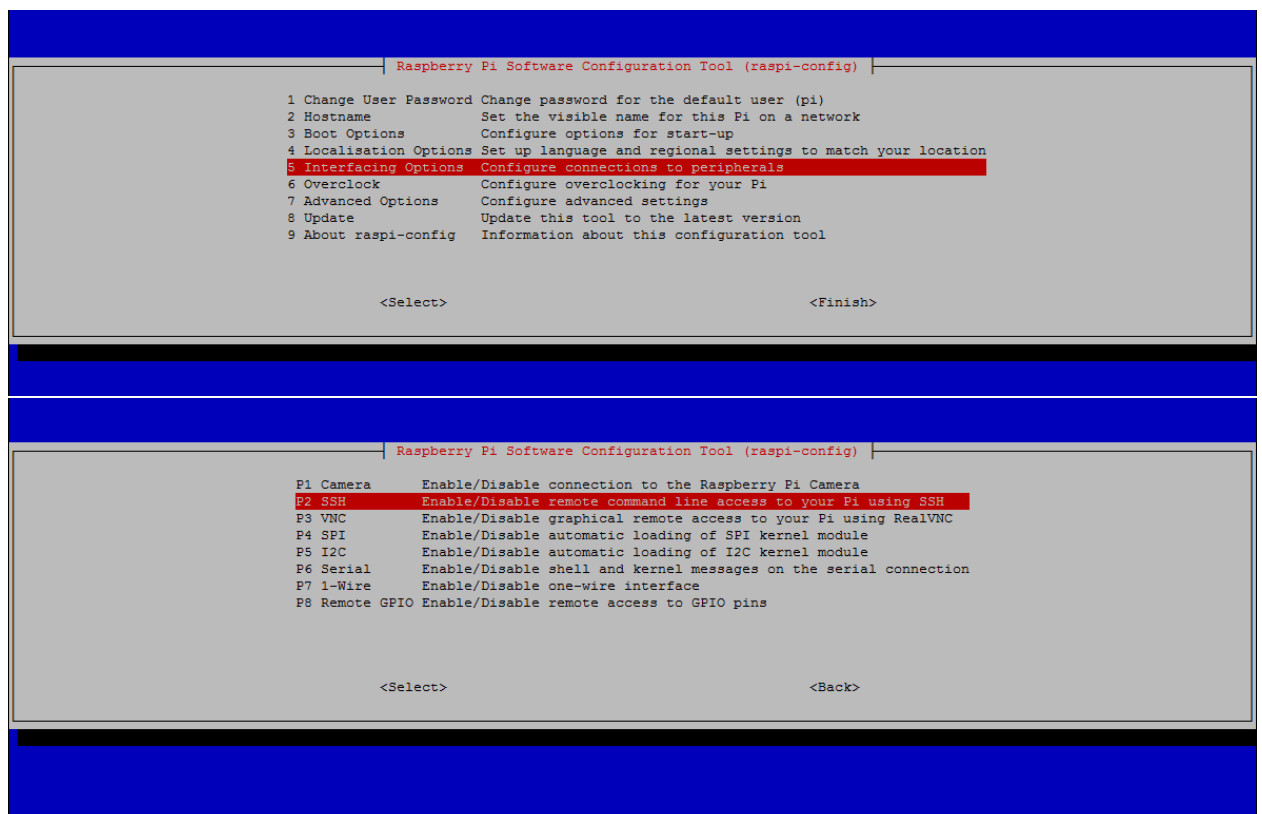
wlan0       Link encap:Ethernet  HWaddr b8:27:eb:1a:58:6e
            inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::f4c1:1cd7:8d67:991e/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:2424 errors:0 dropped:0 overruns:0 frame:0
            TX packets:2506 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:187477 (183.0 KiB)  TX bytes:916957 (895.4 KiB)

root@raspberrypi:/#
```

IP-адрес в строке wlan0 – как раз есть ваше (в данном случае моё) подключение к Wi-Fi.

Следующим шагом надо настроить удалённое подключение к вашей RPi Zero по SSH. Делается это через встроенное меню, командой:

```
sudo raspi-config
```



Сохраняемся и перезагружаемся.

На этом этапе можно отключить клавиатуру и подключить RTL-SDR приёмник к освободившемуся USB-порту. Все настройки теперь можно делать с удалённого компьютера. Для этого подходит программа PuTTY. Настройте соединение на IP-адрес малинки-зеро и подключитесь к ней.

Дальше процедуры точно такие же, как и при настройке сервера на обычном компьютере. Надо сделать полный апдейт-апгрейд системы:

```
sudo apt-get update && sudo apt-get upgrade
```

Устанавливаем недостающие библиотеки:

```
sudo apt-get install libfftw3-3 libpng12-0
```

Устанавливаем драйвера на RTL-SDR приёмник:

```
sudo apt-get install rtl-sdr
```

Проверяем по необходимости соединение с приёмником. Дальше, необходимо каким-нибудь образом перенести папку websdr в папку /home или /home/pi

Сделать это можно 2я способами:

- А) Скачать папку непосредственно на флешку со стороннего Linux компьютера
- Б) Скачать с интернета или локального компьютера по сети командой WGET. (Гуглим)

Ну, а дальше, всё как на обычном компьютере, только максимальная полоса для Raspberry Pi Zero - 256кГц, а для обычной Raspberry Pi 3 – 1МГц и возможно подключение 2х приёмников. Запуск сервера производится командой `./websdr-rpi`

ЗАКЛЮЧЕНИЕ

В статье не рассмотрены методы настроек сервера в режиме автоматического автозапуска при пропадании питания и варианты хостинга вашего сайта с вашим websdr-приёмником. Эти вопросы выходят за рамки данной статьи и подразумевают наличие углубленных знаний в ОС Linux + к тому от версии к версии, методы автозапуска процессов сильно изменяются. Хостинг сайта – это, вообще, отдельная тема специфического опыта. Изучайте, пробуйте, поднимайте свои websdr-сервера!

Больше WEBSDR приёмников, хороших, красивых и на разные диапазоны!!!

Если вы всё прочитали и вам стало интересно попробовать поднять WEBSDR-сервер у себя, то...

[СКАЧАТЬ СЕРВЕР С Я-ДИСКА V1](#)

[СКАЧАТЬ СЕРВЕР С Я-ДИСКА V2](#)