Представляю Вашему вниманию программу приема сигналов OP-32.

Автор: RN3AUS/Alex

1. Общие положения.
Внимание:
- программа не претендует быть заменой "стандартной" Opera v1.5.x by EA5HVK;
- программа не преследует целью превзойти качество обнаружения слабых сигналов OPDS by DF6NM.

Автор выражает признательность Jose EA5HVK и G0NBD за разработку и реализацию превосходного режима Opera.
Автор выражает признательность Markus DF6NM за разработку и предоставление открытых исходных кодов превосходной программы OPDS.
Некоторые идеи и программные решения заимствованы из opds2h5c.bas

Автору неизвестно, как устроена программа Opera v1.5.x by EA5HVK. Все возможные алгоритмические совпадения непреднамеренны и вполне естественны.

Автором сделана попытка реализовать мягкое списочное декодирование максимального правдоподобия, снабдив программу дополнительными удобными опциями. Автор надеется, что программа будет полезна в тех случаях, когда по каким-либо причинам затруднительно использовать Opera v1.5.x by EA5HVK или OPDS by DF6NM.

Программа предлагается по принципу «как есть» без каких-либо гарантий. Исходные коды декодера открыты и могут использоваться всеми желающими.

2. Основные сведения о программе.
Программа предлагается в двух вариантах:
- консольном Op32RxC.exe (удобно использовать для грабберов)
- GUI Op32Rx.exe (для наблюдений и экспериментов)

Обработка сигнала в обеих версиях происходит одинаково.
В графической (GUI) версии дополнительно поддерживается сохранение скриншотов.

Что делает программа.
Программа обрабатывает отсчеты сигнала со звуковой карты или из wav-файла, демодулирует их и пытается декодировать сигнал в формате Opera-32. Если декодер обнаружил совпадение контрольной суммы, декод записывается в текстовый файл, а также отправляется рапорт на pskreporter. Программа имеет встроенный автоматический подавитель импульсных помех (NB).

3. Чем Op32Rx отличается от "стандартной" Opera v1.5.x by EA5HVK:
- имеет открытый исходный код на C++ (CBuilder);
- работает с произвольным Dial 500-4500 Гц;
- кроме отправки спотов на сайт, сохраняет декоды в текстовом файле (как detected.txt в opds);
- использует список частот для глубокого поиска сигнала, этот список автоматически пополняется при каждом успешном декодировании;
- можно задать список игнорируемых частот (например, линий Loran);
- правильность декодов может проверяться по списку известных корреспондентов (как callsloc.txt в opds);
- может декодировать сигнал из wav-файла;
- реализован вариант работы декодера со списочным декодированием (с коротким и длинным списком, соответственно глубокий и сверхглубокий поиск). Это позволяет правильно декодировать очень слабые или замирающие сигналы, что не всегда удается сделать стандартным декодером. (Впрочем, автору неизвестно, как реализован декодер в оригинальной программе Opera v1.5.x by EA5HVK; возможно, там это сделано лучше.)
- отображает спектр с длительным накоплением;
- позволяет автоматически сохранять скриншот водопада спектра в момент декодирования с отметкой на скриншоте трассы сигнала;
- расходует мало памяти (5-9 MB);
- не использует никакой дополнительной информации, поступающей не из эфира;
- имеет определенное число настроек, что позволяет оптимизировать использование ресурсов компьютера;
- возможно, способна обнаружить более слабые сигналы (в ходе экспериментов бывали такие случаи, что OpRx обнаруживала сигнал, а Opera v1.5.4 нет; при этом обратные ситуации не наблюдались).

4. Чем Op32Rx отличается от OPDS by DF6NM:
- работает автономно, не нуждается в сторонних программах обработки сигнала со звуковой карты (SpectrumLab);
- можно в режиме offline обрабатывать предварительно записанные wav-файлы;
- осуществляет настоящее декодирование, а не корреляционный выбор наиболее похожей копии известного сигнала;
- может декодировать любой не известный заранее позывной;
- отправляет споты в Интернет;
- можно назначить список частот для глубокого поиска сигнала, этот список автоматически пополняется при каждом успешном декодировании;
- в случае двухкратного декодирования позывного, отсутствующего в листе callsloc.txt, автоматически запоминает его в файле known_list.txt;
- отслеживает медленный дрейф и блуждания частоты передатчика/приемника без ухудшения качества декодирования. Бывали случаи, когда частота настройки приемника испытывала температурный дрейф на 0,2 Гц за время передачи. При этом сигнал декодировался

без потери качества.
- отслеживает различие тактовых частот передающей и приемной сторон.

5. Как это работает (технические подробности).
Сигнал берется со звуковой карты. Если таких устройств несколько, есть возможность выбрать нужное.
Частота дискретизации 12 кГц, 16 бит.
В настройках можно выбрать центральную частоту НЧ сигнала, соответствующую центру полосы 137500 Гц. Например, при Dial=136 кГц центральная частота спектра OP-32 сегмента будет равна Fc=1500 Гц. Есть возможность установить Fc от 500 Гц до 4500 Гц. Это делается в ini-файле. (Далее мы рассмотрим все настройки в этом файле подробно.)
Определяется средний уровень сигнала. Если присутствуют импульсы, превышающие утроенное значение среднего уровня, импульс подавляется (NoiceBlanker).
Далее очищенный от импульсных помех сигнал фильтруется цифровым КИХ-фильтром с полосой 200 Гц по уровню -3 дБ.
Для трех номиналов "стандартных" Dial (135000 Гц, 135500 Гц и 136000 Гц, с возможностью точной подстройки в пределах плюс-минус 10 Гц) применяются заранее рассчитанные фильтры Баттерворта 6 порядка с 255 отводами, имеющие хорошую равномерность в полосе пропускание и большое подавление за ее пределами.
Если же Dial нестандартный, то КИХ фильтр рассчитывается средствами программы (методом оконного взвешивания с окном Блэкмана-Харриса). Такой фильтра имеет заметно большую неравномерность в полосе пропускания. Количество "отводов" определяется автоматически и не превышает 1024 (в реальности получается около 270).
Отфильтрованный сигнал гетеродинируется и переносится по частоте вниз на центральную частоту 150 Гц (весь спектр лежит в пределах от 100 до 200 Гц).
Для подавления побочных продуктов преобразовния служит еще один КИХ-фильтр с 64 отводами.
Далее осуществляется первая децимация в 24 раза, теперь отсчеты следуют с частотой 500 раз в секунду.
Для осуществления узкополосной согласованной фильтрации в программе применяется линейка из цифровых рекурсивных БИХ фильтров 2-го порядка (цифровой колебательный контур) с полосой пропускания 0.122 Гц по уровню -3дБ (что равно 1/8.192 с - обратная величина длительности одной посылки). Количество фильтров в линейке 1230, что несколько избыточно: в принципе, достаточно иметь 100 / 0.122 = 820 фильтров для перекрытия всего диапазона 100 Гц. Однако эксперименты показали, что чувствительность приема возрастает, если фильтры "стоят" несколько плотнее; в результате была выбрано полуторное количество фильтров.
Может показаться странным применение "цифровых колебательных

контуров" там, где обычно применяют быстрое преобразование Фурье. Возможно, вычислительная эффективность была бы несколько выше, но так как на фильтры поступают отсчеты после децимации, то выигрыш был бы не велик. С другой стороны, наглядность при использовании фильтров получается выше, можно экспериментировать с детектированием и количеством фильтров, чего мы были бы лишены при FFT. Кроме того, FFT уже «набило оскомину»; гораздо интереснее пойти нестандартным путем.

После узкополосных фильтров стоят детекторы. Это может быть либо простой квадратичный амплитудный детектор (когда входной отсчет фильтра перемножается сам на себя), либо квазикогерентный (квадратурный) детектор. Эксперименты показали, что оба детектора приблизительно эквивалентны в плане чувствительности, однако когерентный детектор расходует больше вычислительных ресурсов. По умолчанию когерентный детектор отключен. Его преимущество начинает быть заметно при предельно слабых сигналах, за границей обнаруживающей способности декодера; в этой ситуации применение когерентного детектора несмного увеличивает вероятность обнаружить сигнал. Затем сигнал фильтруется ФНЧ (БИХ-фильтр эквивалентный сложной RC-цепочке) с частотой среза обратной длительности посылки (0.123 Гц).

Далее происходит децимация сигналов с выходов детекторов еще в 1024 раза, так что на длительность одной посылки приходится ровно 4 отсчета (около 1 отсчета за 2 секунды).
После фильтрации и детектирования сигналы поступают в буфер памяти, имеющий 1230 каналов и длиной 240 (количество посылок Opera плюс одна) * 4 (количество отстчетов за посылку) = 960 ячеек. Именно этот буфер и занимает основное место в оперативной памяти 1230 * 960 * 4 (размер float в байтах) = 5 МБайт.
Далее обрабатываются только отсчеты из этого буфера, хранящего в себе информацию за последние 32 минуты.
По каждому каналу (под термином канал понимается выход согласованного фильтра) вычисляется текущая сумма, что соответствует энергии сигнала на длительности передачи Op-32 (32 минуты). Эти значения в дальнейшем используются для отображения амплитудного спектра и водопада.
Для уменьшения количества вычислений декодированию подвергаются не все каналы, а лишь те из них, где есть максимум накопленного сигнала. Максимумы ищутся следующим образом. Каждое значение сравнивается с несколькими соседними значениями (количество их задается параметром MIN_PEAK_SPACE). Если соседние значения меньше текущего – то это максимум. Таких максимумов в спектре очень много. Чтобы сократить их количество, применяется следующий алгоритм. Пусть задано желаемое количество максимумов для обработки (параметр N_MAX). Если найдено больше максимумов, чем требуется, то выполняется «прореживание»: весь спектр разбивается на несколько участков, в каждом из них удаляется максимум, имеющий наименьшую

амплитуду. Так повторяется до тех пор, пока не останется требуемое количество максимумов. Разбивка спектра на участки выполняется из следующих соображений. Если входной фильтр имеет большую неравномерность, или присутствует сильный сигнал, то в часть максимумов, лежащая ближе к «горбу» фильтра будет всегда иметь большую амплитуду, чем все остальные (то есть сама «шумовая полка» будет иметь «горб»). Если просто выбирать определенное количество максимумов с наибольшей амплитудой, все они будут сгруппированы либо около мощного сигнала, либо около «горба» фильтра. Это нехорошо, так как наиболее интересные слабые сигналы от далеких корреспондентов останутся без внимания. Применение описанного выше алгоритма «прореживания» максимумов позволяет вести отбор, когда все они находятся в примерно равных условиях.
Обычно достаточно установить параметр N_MAX=30…40. Если же вычислительные ресурсы не очень критичны, то проще указать N_MAX=-1, и программа будет обрабатывать все найденные (даже маленькие) максимумы.
Из числа каналов, выделенных для дальнейшей обработки, исключаются те, частоты которых совпадают с перечнем частот из файла loran_list.txt (если параметр IGNORE_LORAN_LINES=1).
Если указано USE_FREQUENCY_LIST=1, то на обработку берутся каналы, соответствующие частотам из файла freq_list.txt, независимо, есть ли на них спектральные максимумы или нет. Кроме того, для этих каналов включается режим списочного декодирования с длинным списком (Ultra Deep Decoder).

Декодер реализует алгоритм максимального правдоподобия, осуществляя перебор всех вариантов кодовых комбинаций, используемых Opera. Таких комбинаций 17, каждая комбинация имеет 7 вариантов. (Более подробно о структуре и способе формирования сигнала Opera можно прочитать в файле opera_protocol.pdf). Вычисляются кооreляционные суммы, из них выбираются наибольшие, а также вторые после них. Формируется принятый информационный вектор, имеющий наибольшую корреляционную сумму, а также список его вариантов, с заменой части бит на менее вероятные (из вторых по корреляции кодовых комбинаций). Первый список состоит из 19 вариантов (1 самый вероятный + 17 вариантов с заменой 1 кодовой комбинации на менее вероятную + 1 вариант с заменой всех кодовых комбинаций). Второй список составляется из замен уже двух кодовых комбинаций одновременно и состоит из 136 вариантов. Можно было бы строить и еще более длинные списки, но при этом начинается резко увеличиваться количество ложных декодирований.
Отметим, что декодер учитывает возможность некоторого расхождения тактовых частот приемной и передающей сторон, так что суммарная длительность сигнала может быть несколько длиннее или короче ожидаемой. Осуществляется поиск в пределах плюс минус длительности одной посылки (8.192 с), что соответствует расхождению тактовых частот 1/239=0,42%= 4184 ppm (это довольно много).

Кроме того, декодер осуществляет поиск максимального правдоподобия и в двух соседних частотных каналах (соседних фильтрах), чем частично компенсируется дрейф частоты приемника. Качество декодирования не ухудшится, если уход частоты (в одну сторону) составит за длительность одной передачи (32 мин) не более 0,2 Гц (допустимо и больше, если дрейф плавный); допускается также блуждание частоты около номинала в обе стороны на эту же величину. Если дрейф будет иметь большую величину, то результирующее отношение сигнал шум начнет уменьшаться. В целом для ДВ такое требование к стабильности является достаточно легко реализуемым.
Декодер осуществляет перебор вариантов до тех пор, пока список не закончится, либо не произойдет совпадение контрольной суммы декодированной информации (позывной, передаваемой в Opera, защищается контрольной суммой CRC-16 (16 проверочных бит) и второй контрольной суммой CRC-16, из которой передается 3 бита). Если совпали обе контрольные суммы, то декодер возвращает декодированный позывной. Если совпала только одна (внутренняя 16 бит) контрольная сумма, то декодер помечает этот позывной как недостоверный (со знаком вопроса). Этот декод принимается во внимание, если при этом соотношение сигнал-шум на выходе фильтра превысило некоторый порог (OP_THRESHOLD=4 – порог равен 4 дБ).
Что представляет собой это отношение сигнал-шум. Для наиболее совпадающего варианта сигнала имеется корреляционная сумма – это энергия нашего сигнала. Одновременно мы имеем величину энергии сигнале в этом частотном канале – это есть энергия сигнала + удвоенная энергия шума (шум есть и во время посылок и в паузах между ними). Отсюда находим отношение сигнал шум Eb/No=10*log(2.0*Psn/(Psn-Ps)), где Psn – мощность смеси сигнала и шума, Ps – мощность сигнала (корреляционная сумма декодера). Численные значения Eb/No по результатам эксперимента лежат в диапазоне от +4.0 дБ и выше; например, сигнал, удовлетворительно видный на спектре и дающий декод в «стандартной» Opera -36 dB (opds -37.1dbOp) имеет Eb/No +4.8 dB. Слабые сигналы, которые можно декодировать уже на самой грани возможностей глубокого декодера и которые уже не всегда декодируются Opera, имеют Eb/No около +4 dB. Иногда случались правильные декоды и с Eb/No чуть ниже +4 dB, однако при этом идет множество ложных декодов (но однако с правильными контрольными суммами!). Поэтому порог для «подозрительных» декодов разумно установить примерно 4 dB).
Отдельно нужно оговориться, что, возможно, не совсем правомерно использовать здесь термин «Eb/No» - нормированное битовое отношение сигнал-шум. Однако кажется, по смыслу измеренная нами величина близка к смыслу термина «Eb/No»; кроме того, термин достаточно краток и удобен для отображения на экране, поэтому я решил воспользоваться этим обозначением и в дальнейшем.
Итак, на выходе декодера мы имеем декодированный позывной, с правильными контрольными суммами (суммой). Однако это еще не значит, что он действительно присутствовал в эфире. Практика показывает, что ложное обнаружение по шумам произвольных позывных с

совпадающими CRC происходит довольно часто, порядка одного раза в полчаса/час. Можно было бы защититься от ложных декодов, введя сравнение с некоторым порогом и не регистрировать декод, если Eb/No ниже этого порога. Однако не так редки случаи, когда Eb/No правильного декода на слабом сигнале бывал ниже Eb/No ложного декода! Нас же интересует прием наиболее слабых сигналов; не хотелось бы потерять их, установив порог таким образом, чтобы надежно не пропускать ложные декоды. Возникает классическая дилемма: либо пропуск сигнала, либо ложная тревога. Кроме того, на практике часто получается следующая ситуация: если в эфире присутствует чей-то сильный сигнал, то на нем при ультра глубоком поиске легко «набираются» ложные декоды с очень хорошим уровнем Eb/No, так что сравнение с порогом здесь, к сожалению, никак не поможет.
В программе реализован иной метод защиты от ложных тревог. Нужно согласиться, что в ДВ эфире работает очень немного корреспондентов, их позывные известны, появление каждого нового корреспондента не проходит незамеченным. Для программы OPDS by DF6NM используется файл callsloc.txt, где указаны позывные и локаторы всех реально работающих или когда-то работавших корреспондентов. Поэтому для «отсеивания» ложных декодов (а это всегда очень своеобразные позывные!) используется поиск декодированного позывного в файле callsloc.txt. Если он там есть, значит декод с большой долей вероятности правильный. Заодно и определяем расстояние до корреспондента.
Однако, вдруг в эфире присутствует новый корреспондент, которого пока еще нет в callsloc.txt? Было бы обидно отбросить этот правильный декод только лишь по причине отсутствия позывного в файле. Для этих случаев предусмотрен механизм, когда дважды принятый произвольный позывной считается отныне «известным» и он помещается в файл known_list.txt. Только локатор остается нам не известен. Мы можем, увидев декод от такого корреспондента, поместить его позывной отныне в callsloc.txt, дополнив qth-локатором, либо, если это ошибка, стереть его из файла known_list.txt.

Для увеличения вероятности правильного декодирования слабых сигналов служит, как уже говорилось, режим списочного декодирования с длинным списком. Однако это загружает CPU. Для достижения компромисса предназначен режим использования файла предпочитаемых частот. Этот перечень может быть составлен нами заранее, так как многие из нас передают на одной и той же частоте. Кроме того, если частоты, на которой произошел правильный декод, нет в списке, она будет туда автоматически занесена.

Строка декода, выводимая на экран и записываемая в файл, содержит дату, время, позывной, частоту, QRB (если известен qth), оценку битового отношения сигнал-шум на выходе декодера Eb/No, отношение

сигнал-шум SNR (вычисляется отношение данного спектрального максимума к среднему значению амплитуд спектра по всей полосе 100 Гц; чтобы результат как-то соотносился с результатами Opera, к найденной величине прибавляется поправка, задаваемая параметром OPERA_DB_OFFSET=-10 – для слабых сигналов это примерно -10 дБ). Выводится количество итераций декодера (1 - декодировано сразу правильно, меньше 20 – с коротким списком, больше 19 – с длинным списком). Затем, если совпала только первая CRC, выводится знак '?'. Если декодированный позывной отсутствует в списке известных позывных, выводится знак '-'.
Если разрешена отправка спотов на сайт pskreporter, то в случае успеха в последнем поле выводится знак '+', а неудачи 'x' (в консольной версии; в GUI версии результат выводится в правом поле Status Bar).
Вывод декодов на экран и запись в файл производятся в «инверсном» порядке, то вверху будет самый последний по времени декод. Пожалуйста, имейте в виду, что вставка новой строки в файл всегда осуществляется в позицию третьей от начала файла.

6. Ниже приведена структура ini-файла с комментариями:

```
[OP-32]
мой позывной
MYCALL=RN3AUS
мой локатор
MYQTH=KO85FN
заголовок для файла
MYHEADER=RN3AUS KO85FN Opera-32 decoded (Op32RxC RN3AUS software)

Центральная частота аудио спектра
F_CENTER=1500
Частота настройки приемника
F_DIAL=136000
Минимальное расстояние между спектральными максимумами
(в количестве отсчетов спектра, один шаг равен 0.08 Гц)
MIN_PEAK_SPACE=3
Количество спектральных максимумов ( -1 – авто)
N_MAX=-1

файл для записи спотов
FILE=my_detected.txt

игнорировать частоты из списка loran_lines.txt (1-да 0-нет)
IGNORE_LORAN_LINES=0
всегда анализировать частоты из списка freq_list.txt (1-да 0-нет)
USE_FREQUENCY_LIST=1
отсеивать незнакомые позывные, отсутствующие в callsloc.txt и
known_list.txt
USE_KNOWN_CALLSLIST=1
```

```
использовать декодер с глубоким поиском (короткий список)
DEEP_DECODER=1
использовать декодер с длинным списком
ULTRA_DEEP_DECODER=0
использовать когерентный детектор (0-амплитудный квадратичный)
USE_COHERENT_DETECTOR=0
поправка оценок SNR (dB) для единообразия с Opera
OPERA_DB_OFFSET=-10
порог (dB) для декодов с несовпавшей внешней 3-х битовой CRC
OP_THRESHOLD=4
разрешить сохранять скриншоты спотов (только для GUI версии)
CAPTURE_ENABLE=1
путь для сохранения скриншотов (только для GUI версии)
CAPTURE_PATH=Capture\

[SOUND]
используемая звуковая карта (-1 по умолчанию)
DEVICE=-1

[PSKREPORTER]
HOSTNAME=report.pskreporter.info
PORT=4739
разрешить отправлять споты на сервер
UPLOAD_SPOT=1
```

7. Установка программы и начало работы.

Программа не требует установки, достаточно просто поместить следующие файлы в любое место на диске:

- Op32Rx.exe (Op32RxC.exe) – исполняемый файл
- Op32Rx.ini (Op32RxC.ini) – конфигурационный ini-файл. Если он отстуствует, он будет создан автоматически со значениями по умолчанию.
- callsloc.txt – файл с позывными и локаторами, полностью совпадает по структуре и смыслу с одноименным файлом для OPDS by DF6NM.
- freq_list.txt – файл с частотами, которые будут постоянно анализироваться, независимо от наличия спектральных пиков. Если файл отсутствует, он будет создан автоматически. Файл заполняется также автоматически по мере появления декодов. Частоты указываются в Гц в формате с плавающей точкой, например 137531.503
- loran_lines.txt – файл с перечнем игнорируемых частот. Если частота указана одновременно в freq_list.txt и в loran_lines.txt, то сигнал на этой частоте будет анализироваться. Файл заполняется вручную. Частоты указываются в Гц в формате с плавающей точкой, например 137531.503
- known_list.txt – файл со списком не менее двух раз декодированных позывных, отсутствующих в файле callsloc.txt.
- PSKReporter.dll – динамическая библиотека, необходимая для отправки спотов в Интернет. Без этой библиотеки программа не

запустится.

При первом запуске программа предложит (если ini-файл отсутствовал) указать Ваши позывной и локатор в ini-файле. Не забудьте указать их и в заголовке MYHEADER вместо автоматически вставленных значений <null>.

В процессе работы иногда просматривайте файл known_list.txt чтобы удалить из него ложные позывные, если таковые в нем есть, либо чтобы перенести позывной нового корреспондента в файл callsloc.txt (на что хотелось бы надеться – увы, новые корреспонденты на диапазоне 136 кГц появляются не чаще раза в год ☹).

8. Рекомендуемые поараметры.

а) Если Ваш компьютер достаточно быстрый, то лучше установить следующие параметры:

```
MIN_PEAK_SPACE=3
N_MAX=-1
USE_FREQUENCY_LIST=1
USE_KNOWN_CALLSLIST=1
DEEP_DECODER=1
ULTRA_DEEP_DECODER=1
USE_COHERENT_DETECTOR=1
```

Это обеспечит максимально глубокий поиск сигналов.

б) Если же компьютер не очень быстрый, то для экономии ресурсов процессора можно установить параметры:

```
MIN_PEAK_SPACE=3
N_MAX=30
USE_FREQUENCY_LIST=1
USE_KNOWN_CALLSLIST=1
DEEP_DECODER=1
ULTRA_DEEP_DECODER=0
USE_COHERENT_DETECTOR=0
```

Такие настройки обеспечивают достаточную, близкую к предельной, чувствительность. На частотах из freq_list.txt будет происходить сверхглубокий поиск. С этими настройками требования к производительности CPU такие же, как у Opera v1.5.x by EA5HVK, а памяти расходуется во много раз меньше.

в) наконец, если у Вас старый компьютер и производительности не хватает, попробуйте

```
MIN_PEAK_SPACE=6
N_MAX=20 (или даже 10)
IGNORE_LORAN_LINES=1
USE_FREQUENCY_LIST=0
```

USE_KNOWN_CALLSLIST=1
DEEP_DECODER=0
ULTRA_DEEP_DECODER=0
USE_COHERENT_DETECTOR=0
В этом случае и CPU и RAM будут загружены минимально. Возможен пропуск слабого сигнала, если присутствует несколько сильных сигналов и/или Лорановских линий. Очень удобно игнорировать лорановские пики, внеся их в список; используйте также список ожидаемых частот freq_list.txt (и параметр USE_FREQUENCY_LIST=1), если ожидается работа корреспондентов на своих обычных частотах.

Внимание: основное влияние на CPU оказывают параметры N_MAX, ULTRA_DEEP_DECODER и USE_COHERENT_DETECTOR.

Внимание: использование параметра USE_KNOWN_CALLSLIST=1 следует считать обязательным для обеспечения отсутствия ложных декодов (это основное средство защиты от них!). Отключать фильтрацию декодов по списку известных позывных можно для экспериментов, при этом не забудьте отключить (UPLOAD_SPOT=1) и выгрузку спотов в Интернет!

9. Особенности GUI-версии:
- занесение частот в ожидаемые или игнорируемые списки можно выполнять, наведя указатель мыши на водопад или спектр и нажав правую кнопку мыши.
- щелчок левой кнопкой мыши на водопаде или спектре (если Вы увидели нечто похожее на сигнал OP-32), включает отображение выходного сигнала согласованного фильтра и включает режим обязательного поиска/декодирования на этой частоте. Для отмены сделайте двойной клик на осциллограмме (верхняя картинка).
- на спектре и водопаде отображается линейный (не логарифмический) масштаб амплитуд. Выполняется автоматическое масштабирование к максимуму. Так сделано для упрощения интерфейса программы с сохранением достаточной наглядности. Автор не ставил перед собой задачи повторить или хотя бы приблизиться к качеству превосходного спектроанализатора SpectrumLab by DL4YHF.
- в этой версии 1.0.0 введен пока что экспериментальный режим корреляционного детектирования сигнала (по аналогии с OPDS). Это может порождать множество ложных декодов, так как вопрос выбора правильного порога OP_THRESHOLD=4 пока что остается не совсем ясным. До тех пор, пока от момента запуска программы прошло менее 32 минут, пожалуйста, не ставьте галочку «Correlation OPDS»! Пока внутренний буфер не наполнился, могут появляться ложные декоды с большим Eb/No. После истечения указанного времени, можно включить этот режим. В будущих версиях программы я планирую что-то усовершенствовать ☺, либо отказаться от этого режима, если он не даст прибавки эффективности поиска сигнала.

10. Особенности консольной версии.

В командной строке можно указать файл для анализа:
Op32RxC.exe myfile.wav
Файл должен быть записан в формате 12 ksps, 16 bps. Программа запустится и начнет анализировать файл, о чем будет свидетельствовать надпись “processing file…”. По окончании анализа программа автоматически перейдет в режим обработки сигналов со звуковой карты, о чем будет свидетельствовать пропадание вышеприведенной надписи. Время анализа записи продолжительностью 35 минут (48 Мбайт) составляет порядка 5 минут и сильно зависит от настроек декодера и производительности компьютера.
Консольная версия предназначена для использования на грабберах, где нет необходимости использовать графический пользовательский интерфейс для текущих настроек. Кроме того, разобраться в исходных кодах консольной версии несколько проще, как и, возможно, портировать его на другие платформы.

Выявленные недостатки, равно как и Ваши пожелания, желательно сообщать автору: rn3aus@mail.ru

С уважением, RN3AUS/Alex
73!