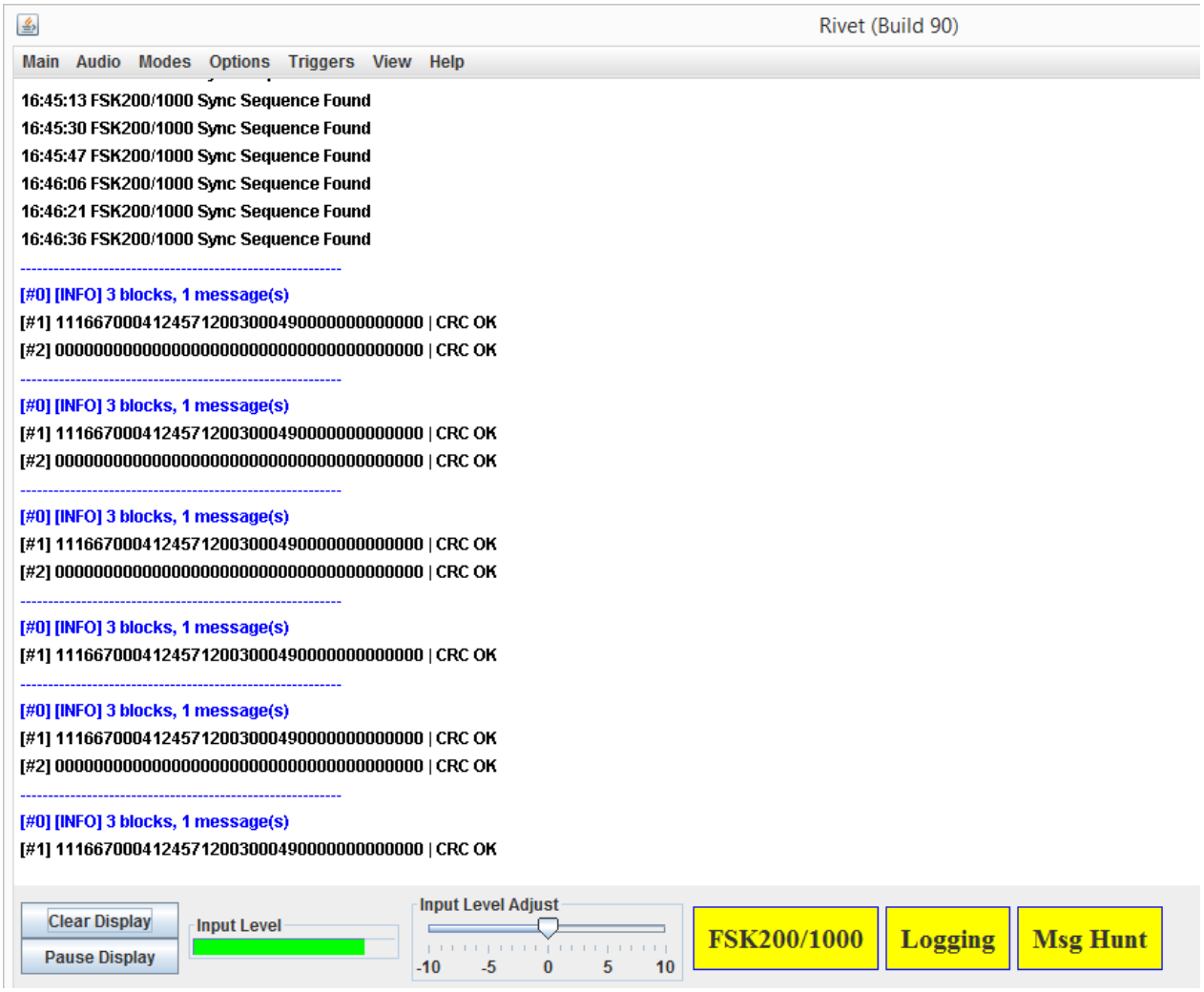
Rivet manual

Rivet is a popular free decoder created by Ian Wraith. This manual is derived from info from the Rivet website plus some additional info. Compiled for UDXF and Numbers & Oddities by Ary Boender.



Rivet is written by Ian Wraith
Rivet build #90 : the FSK 200/1000 and XPA2 code is improved by Daniel Ekmann
Download at: http://www.signalshed.com/rivet/index.html

A word from the author:
Rivet is a open source decoder written in Java which decodes various HF data modes that have either been forgotten by the better known decoder programs or which can only be decoded by expensive programs that are beyond the reach of the average hobbyist. The program doesn't need any other hardware to operate all it requires is a sound card with an audio feed from HF receiver.

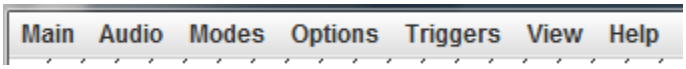
Those of you who have used decoder programs for HF data modes before will find Rivet a little different. Firstly Rivet doesn't need you to precisely tune your receiver to signal for it to be decoded. Instead Rivet takes advantage of a modern PCs processing power and adjusts itself to the incoming signal. This means you don't waste vital minutes messing around with a tuning indicator when you want to see what is being transmitted. The downside is that Rivet does need to run on a fairly powerful PC with at least 2 cores. Secondly Rivet is designed not only to accept a direct sound input from a sound card but it can also decode data from a .WAV file (which must be mono and have a sample rate of 8 KHz). So you can

record a transmission while you are searching through the HF spectrum and record it later. If you click here you will find several recordings you can use to test the program.

Lastly Rivet is a free and open source program something which is very unusual for a HF data mode decoder. To make it better I need your input so if you find any bugs or have any feature suggestions please email me. If you are a programmer then use Github to fork Rivet and add some new features yourself then if I like them I will add them to the program.

Ian Wraith

Menus



Main

- **Copy All to the Clipboard**
  The entire contents of the screen display are copied into your clipboard. Useful if you wish to include a section of a decode from Rivet in an email or word processor document.
- **Load a WAV File**
  Rivet not only decodes audio directly from your radio but can also decode data contained in WAV files which were recorded previously. Note that Rivet can only decode WAV files which were recorded in mono mode and at certain sample rates (usually 8000 Hz).
- **Reset Decoding State**
  If this menu item is selected then Rivet resets to the default settings for the particular mode you have selected. So if Rivet has set its levels and symbol timing on a signal and this option is selected then Rivet will reset and try to reacquire levels and symbol timing.
- **Save the Current Settings**
  If you click on this all of the settings you have selected (Mode etc) will be saved in a file called "rivet_settings.xml". From then onwards when you restart Rivet the program will load these settings and use them as its default. To change this select the settings you would like to be your new default and click on this item again.
- **Save to File**
  When this option is enabled all traffic that is decoded by Rivet will also be saved to an ASCII text file. When you select this option a dialog box will appear requesting you give a name and a location for this file.
- **Save Bit Stream to File**
  When you enable this option the program will ask you to select a file name and folder for the file. Afterwards the raw binary output from the program will be a saved in that file (which has a .bsf prefix). It is then possible for you to analyse the data in this file for the purpose of reverse engineering a mode. Please note that this option only currently works when decoding selected modes.
- **Soundcard Input**
  When this is enabled Rivet will accept and attempt to decode audio from the audio source selected by your PCs mixer. I would recommend that you use your PCs Line Input source for this.
- **Exit**
  Use this to shut down the program.

Modes

Use this menu to select the mode you wish to decode.

- CCIR493-4 : A HF selective calling mode
- CIS36-50 (50 baud only currently) : Used by the Russian Navy.
- CROWD36 : Used for Russian diplomatic and intelligence messages
- FSK200/500 : Used for Russian diplomatic and intelligence messages
- FSK200/1000 : Used for Russian diplomatic and intelligence messages
- FSK (raw) : For advanced users to investigate unknown FSK modes
- GW FSK (100 baud) : A commercial ship to shore data system.
- XPA (10 and 20 baud) : Believed used for Russian intelligence messages.
- XPA2 : Believed used for Russian intelligence messages.

Options

- **Baudot & FSK options**
  Here you can enter several options for Baudot & FSK modes
- **Debug Mode**
  When this mode is enabled Rivet will display debugging and diagnostic information about the signal it is decoding. Unless you are a developer I wouldn't recommend you enable this option.
- **Invert**
  When enabled the data received by Rivet will be inverted. So for example if a FSK (Frequency Shift Keying) mode is being received and the high tone normally represents a 1 and a low tone a 0 , if this option is enabled then the high tone will represent a 0 and the low tone a 1. In some modes (e.g CCIR493-4) Rivet will automatically decide if Invert needs to be enabled or disabled and will change it accordingly.
- **Set the CROWD36 High Sync Tone**
  This option is used when you are in CROWD36 mode to tell Rivet the tone number of the high sync tone being used by the CROWD36 station you are monitoring.
- **CIS36-50 options**
  Here you can enter several options for the CIS36-50 mode.

Triggers

See appendix.

View

- **View GW Free Channel Markers.**
  If you are decoding the shore side of GW 100 baud FSK channel you will see a free channel marker decoded every few seconds. This will soon fill Rivets display and may become annoying. If you disable this option the program will ignore free channel markers and you will only see other traffic.
- **Display possible bad data**
- **Display UTC time**
- **Clear display**

Help

- **About Rivet**
  Informs you of the version of Rivet you are using and some information about the author.
- **Enigma2000**
  This option takes you to a web page giving information about the Enigma2000 group.
- **UDXF**
  This option takes you to the UDXF website.
- **Follow Rivet Progress on Twitter**
  This takes you to Rivets authors Twitter page. Follow me to find out the latest news on Rivet.
- **Help**
  This takes you to the Rivet help pages.
- **Sound Sample Files**
  Takes you to a web page where you can download sample sound files to decode using Rivet.
- **More Information on the Modes Decoded By Rivet**
  - CCIR493-4
  - CIS36-50
  - CROWD36
  - FSK200/500
  - FSK200/1000
  - GW 100 Baud FSK
  - XPA
  - XPA2

CCIR493-4

- This mode is used for individual and group selective calling on the HF bands. The data consists of short bursts of 100 baud FSK (Frequency Shift Keying) with a shift of 170 Hz. Rivet decodes this data and displays it so it looks like this:
  12:53:51 CCIR493-4 Individual Selective Call Station 2522 Calling 2519 (Routine)
- CCIR493-4 data can either be decoded live from a HF radio connected to your soundcards line in input or from a prerecorded WAV file. This WAV file however must be recorded in mono mode with a 8000 Hz sample rate.
- To decode this mode tune your radio to the frequency where you know CCIR493-4 is used with the receiver set to either USB or LSB modes. Rivet will wait for the brief CCIR493-4 synchronisation sequence and adapt itself from that then decide if the signal needs to be inverted or non-inverted automatically.

CIS36 50

- This mode can be found all over the HF bands. It is believed to be a Russian Naval mode which transmits FSK (Frequency Shift Keying) at a data rate of 50 baud with a 200 Hz shift and is sometimes known as BEE. Some stations transmitting this mode transmit constantly sending idles between messages while others only transmit when a message is due at fixed intervals. All the CIS36-50 transmissions decoded so far have been encrypted and look like this:
  13:19:29 Message Start
  Sync 0x1414bebe64c
  Session Key is 0x18 0x5 0x1f 0x16 0x4 0x9 0x8 0x1 0x4 0x10
  ZHGANENXGWJBAQEYEXRGBZEN
  End of Message (44 characters in this message 0 of these contained errors)
- Rivet can decode CIS36-50 transmissions live from a HF radio (in USB mode) via the line in input on your computers soundcard or from a pre-recorded WAV (mono and 8000 Hz sample rate only). Rivet calibrates itself from the alternating idling sequence that precedes a CIS36-50 message then looks for a valid sync word and session key before decoding and displaying a message.

- Note that on the LF bands CIS36-50 is transmitted at a speed of 36 baud but at the moment Rivet can't decode this only the 50 baud transmissions.

CROWD36

- CROWD-36 is the name given to a mode used for Russian diplomatic communications by hobbyists. It actual name is believed to be the "SERDOLIK" long range radio-system. It is a MFSK (Multi Frequency Shift Keying) system with 36 possible tones sent a speed of 40 baud.
- Unlike most of Rivets modes CROWD36 decoding isn't automatic and requires the user to to input some information to aid the calibration of the program. All of the CROWD36 broadcasts I have seen so far start with what appears to be a synchronisation sequence made up of four tones as you can see in this spectrogram ..
- CROWD36 synchronisation sequence.
  When it is CROWD36 decoding mode Rivet hunts for four tones (tone#1,tone#2,tone#3,tone#4). Where tone#1 = tone#3 and tone#2 = tone#4 plus tone#1 minus tone#3 is less than 100 Hz. Rivet then calibrates itself on tone#1 which it takes to be the tone number specified by the user. You set this value by clicking on the "Options" menu "Set CROWD36 High Sync Tone" and entering a number between 0 and 33. The broadcasts I have seen either use tone 24 or tone 31 for this value but it does need some experimentation and I would recommend that you record CROWD36 broadcasts (as WAV files , mono and 8000 Hz sample rate) then you can process that file in Rivet trying different values.
- In the past CROWD36 transmitted encrypted information in the form of 5 digit numbers then later moved to online encryption. These days the mode seems to be being used for link setup information with the actual traffic being sent in another mode if the conditions are suitable. So frequently the only decoded traffic is lists of repeating characters such as below.

  AQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAQAFMFMF
  MFMFMCOGISfs1Is22T
  M QLQAQAQAQAQAQAQAQAQAQAQAQAQAQ22YXSQJCTUQfs)79
  EXMQ FQHLNKDCBOYGUIAS10fs21?226?1?:571fs)+9
  EXMQ FQHLNKDCBOYGUIAS10fs21?226?1?:571fs)79
  EXMQ FQHLNKDCBOYGUIAS10fs21?226?1?:571fs)+9
  EXMQ FQHLNKDCBOYGUIAS10fs21?226?1?:571fs)79
- There is a lot of work need doing to Rivet's CROWD36 decoder module and if anyone can provide me with any further information to help I would be very grateful.

FSK 200/1000

- This mode is believed to be used by the Russian intelligence services. Data is transmitted using FSK (Frequency Shift Keying) at a speed of 200 baud with a 1000 Hz shift.

FSK 200/500

- This mode is believed to be used by one of the Russian intelligence services. It transmits data using FSK (Frequency Shift Keying) modulation with the ITA-2 alphabet (with 1.5 stop bits) at a speed of 200 baud with a shift of 500 Hz.
- Rivet can decode FSK200/500 messages directly from a radio in USB mode connected to a soundcards line input or from a WAV file that is mono and with a sample rate of 8000 Hz. The program calibrates itself by looking at the incoming audio and then starts to decode and display the message on the screen. Once a message is decoding do not retune your radio until the message is complete.
  - When there is no message to send in a schedule the line 00000++++++++++162)5761

  will be sent continuously for seven minutes. When there is a message offline encryption is sent which starts like this

  00000++++++++++188=8331
  52281013891141343 =8432
  92874574809254333 =8573
  814347960943078155=8794

- The number to the left of the = sign appears to be the encrypted traffic and consists of 17 or 18 digits. The number to the right is interesting also though. The last digits of this is the line number which you can see incrementing. The messages ends like this:

  28870268039372698 =81275
  423345851935701126=86276
  88837514787689596 =81277
  26158186121423068)57678
- Certain 3 digit codes appear to have special meanings. The ones we have come across so far are ..

  162 Start of a null message
  188 Start of a message
  576 End of message or null

GWK 100Bd FSK

This mode is one of several used by a commercial communications company to contact ships at sea and to send data to them. The mode is defunct now.

XPA

- This is a Russian mode believed to be used by at least one of the Russian intelligence services. It transmits data at either 10 baud (most commonly) or 20 baud using 17 tone MFSK (Multi Frequency Shift Keying).
- Rivet can decode XPA live from a radio (using your soundcards line in input) or from . WAV file recording which is mono and with a sample rate of 11024 KHz or 8000 KHz. The program begins its operation by hunting for the XPA synchronization sequence by looking for four tones (tone #1 , tone #2 , tone #3 and tone #4) it then checks that tones #1 and #3 are the same and that tones #2 and #4 are the same, finally it then checks that there is 760 Hz difference between tones #1 and #2. At that point the program calibrates its timing and the frequency offset of the transmission. Once Rivet has done this do NOT change the tuning of your radio.
- A decoded XPA message looks like this:

  13:13:58 XPA Start Tones Found (correcting by 36 Hz)
  13:14:03 High sync tone found
  13:14:03 Symbol timing found
  Block Sync
  4444444444
  Block Sync
  101 101 101 1 101 101 101 1 101 101 101 1
  4444444444
  Block Sync
  6
  Message Start
  00857 00097 93286 71395 40333 00714 17852 48857 77718 50780 37834 00215 21772 68734 13603
  85646 14094 40723 82696 11038 43675 98847 50732 30954 17592 54577 90765 62197 87165 86007
  68736 35714 24007 86209 96271 89625 35714 19269 81590 60485 02469 82980 59119 82461 77523
  86731 22993 12521 52548 97909 55084 10648 78945 97123 16673 97138 83388 34822 60415 93306
  50071 38438 27154 80657
  Block Sync
  82050 37623 04296 14200 72261 30215 93943 34679 68188 53956 79136 08478 30933 72586 38826
  06910 55823 18377 89051 96813 05847 91457 30993 17025 36671 20034 7380 08659 34217 05873
  30813 11871 61634 20351 39596 7426
  13:14:03 XPA Decode Complete

XPA2

- XPA2 is a Russian mode believed to be used by at least one of the Russian intelligence services. It transmits data at the unusual rate of 7.8 baud using 14 tone MFSK (Multi Frequency Shift Keying).
- Rivet can decode XPA2 live from a radio (using your soundcards line in input) or from . WAV file recording which is mono and with a sample rate of 11024 KHz or 8000 KHz. The program begins its operation by hunting for the XPA synchronization sequence by looking for four tones (tone #1 , tone #2 , tone #3 and tone #4) it then checks that tones #1 and #3 are the same and that tones #2 and #4 are the same, finally it then checks that there is 234 Hz difference between tones #1 and #2. At that point the program calibrates its timing and the frequency offset of the transmission. Once Rivet has done this do NOT alter the tuning of your radio.
- A decoded XPA2 message looks like this:

  13:49:52 XPA2 Start Tones Found (correcting by -1 Hz)
  13:50:29 Symbol timing found
  02435 00146 98804 73855 93507 32951 32279 89275 85524 53888 95942 92198 90557 04769 54888
  20926 94351 53313 31155 63773 55936 08424 26549 01634 43739 83041 11786 65426 09526 69705
  28433 53534 59595 88768 38470 64642 94623 09572 01779 46864 48326 44457 80870 13778 58331
  70218 02211 33066 78858 82209 34788 24458 27877 14600 16550 66300 53399 36220 66529 80022
  84492 18763 77921 74517 06590 40668 78325 11660 78097 35235 66565 05193 08038 23858 02838
  96632 12334 57529 31361 97524 06251 73501 56953 86309 65431 43780 17915 56730 12549 07132
  34694 22306 01444 55806 17712 69823 33014 07770 37248 06682 33638 30834 75725 02321 58679
  91992 44526 72421 05908 33798 91084 28280 23088 32018 74665 75705 58256 00465 27697 04388
  37650 93575 06837 46454 84140 47744 88681 77535 49948 27777 43165 11011 80486 68013 91912
  97794 76189 68125 99885 31533
  13:50:30 XPA2 Decode Complete

Frequently Asked Questions

1. *Why doesn't Rivet have a .exe file ? How do I install and run it ?*
   Rivet is written in a programming language called Java which allows it to run on computers using a variety of operating systems which include MS Windows , Apple OS X and Linux. Instead of having a .exe file it has a .jar file. To run the program you must first install Oracle Java which is free can be downloaded from here. To install Rivet go to the Github download page and click on the link that says "rivet_bxx.jar" where xx is the build number. Save that file into a folder on your PC. Then to run it go to that folder and double click on that .jar file , within a few seconds Rivet should run. A common problem is that Rivet appears to a folder containing lots of files. The usual cause of this is that an unzip type program has decided that it "owns" .jar files rather than Oracle Java.

2. *How do I know my radio is tuned correctly ? and why doesn't Rivet have a tuning aid ?*
   That is where Rivet is different ! In the past decoders require that you tune your radio precisely to the audio frequencies it expects the data to be at. Rivet takes advantage of a modern computers amazing processing power to look at an incoming signal and then it adapts itself to the signal so no tuning aid is needed. This means you can also decode data you recorded in .WAV sound files previously.

   It is important to remember that Rivet adapts itself to a signal in different ways depending on the mode being monitored. With FSK200/500 , FSK200/1000 and GW 100 Baud FSK the program simply looks for the two frequencies separated by the correct bandwidth. With the MFSK modes things are more complex. So with XPA and XPA2 Rivet needs to see the modes start of message sequence to calibrate itself.

3. *Why doesn't Rivet decode this mode ?*
   I probably just haven't got around to adding it ! Working on Rivet is a hobby of mine and sadly I never have as much time as I would like to work on it. If you know of a mode you would like Rivet to decode please contact me and tell me about it.

4. *How could I help improve Rivet ?*
   I am looking for three kinds of people ..

5. *If you are a Java programmer* and are interested in helping please use Github to fork this project and feel free to add any modes or make any changes you want , it is a fully Open Source project. However I would be very grateful if you could send me your changes so I can include them in Rivet and everyone can take advantage of them. I will admit that Rivets documentation needs a lot of work to improve it. If you want to help with that please get in touch.

6. *Rivet doesn't start.* If you doubleclick on a jar file, and your Java application does not start, your .jar association has been hijacked. You can fix the problem with Jarfix.
   http://johann.loefflmann.net/en/software/jarfix/index.html

7. *More Java problems. Error: "Could not create the java virtual machine"*
   If Jarfix doesn't help, try this:
   Go to the Windows config screen; then System; then Advanced
   Go to Environmental Variables
   Add a new variable:
   Variable name: _JAVA_OPTIONS
   Variable value: -Xmx512M

   This specifies the maximum size, in bytes, of the memory allocation pool.

Appendix

An introduction to the Rivet Trigger feature

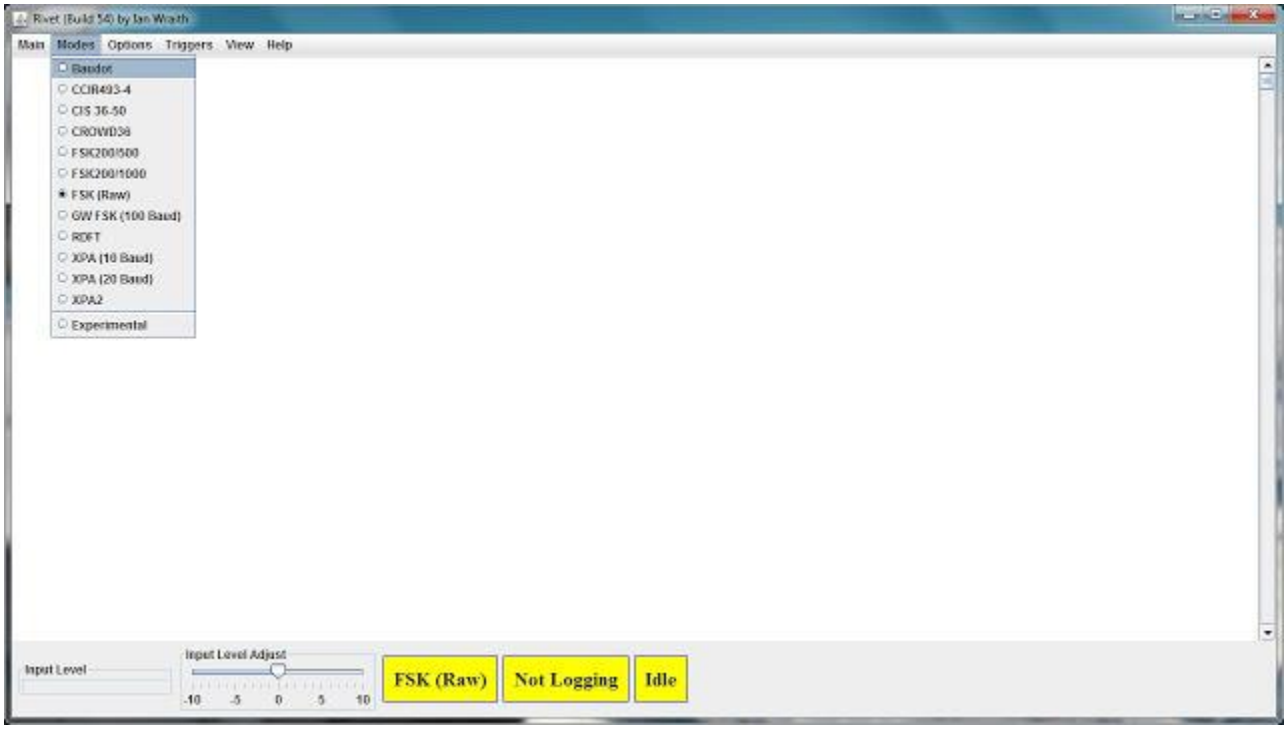
Rivet build 90 allows the user to add , edit and delete Triggers from within the program. Previously users had to manually edit a XML file using a text editor which I realise isn't easy if you don't have any experience with that sort of thing. Since using Triggers is now within anyone's grasp I thought I would write a little step by step tutorial on how to use it.

Firstly I had better explain what Triggers are. Rivet has a mode called **FSK (Raw)** which is intended for users who wish to investigate unknown or little known FSK modes. When selected and after the correct baud rate and shift have been set Rivet displays what the station is transmitting as binary. This is useful when you are initially investigating a mode but soon the amount of data becomes rather overwhelming. To make this mode more useful one of Rivet's users suggested a feature we decided to call a Trigger. Now a Trigger is nothing more than a binary sequence which Rivet stores in its memory and constantly compares with incoming raw FSK data. When the incoming data matches the binary sequence in the trigger then the program does *something* where the something depends on the type of trigger. So far there are three types of trigger ..

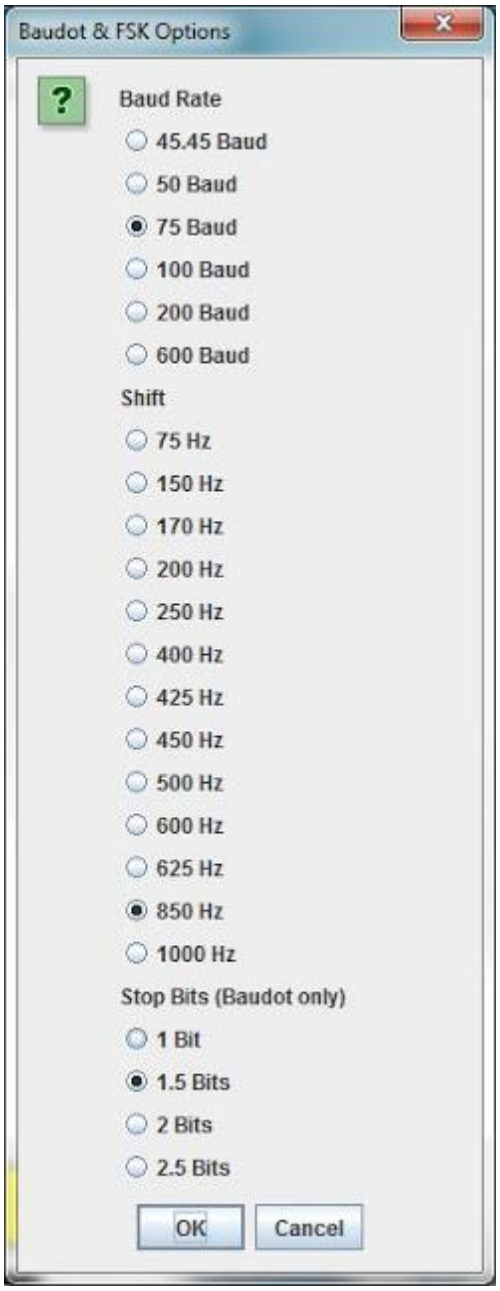
- **Start** trigger. With this type of trigger Rivet displays nothing until it receives the sequence of data defined in the start trigger. At which point the program displays the name of the trigger that has been activated and then displays all incoming data.
- **End** trigger. When Rivet matches the sequence of data in this type of trigger with the incoming data it displays the name of the trigger and then stops displaying incoming data.
- **Grab** trigger. With this type of Trigger when Rivet matches the incoming data with the trigger sequence then it displays the name of the Trigger , the previous *backward grab* number of bits (a number defined by the user) which were transmitted before the trigger sequence and the *forward grab* number of bits (which again has been defined by the user) which follow the trigger sequence. This type of trigger is very useful for packets of data where the synchronisation sequence is in the middle of the packet.

So that is what triggers are now let me show you how to use them. For this example we will have a look at a 75 baud 850 Hz shift synchronous FSK signal using KG-84 encryption most likely sent by the armed forces of a NATO member country.

To use it start up Rivet then select **Raw (FSK)** as the decoder mode.



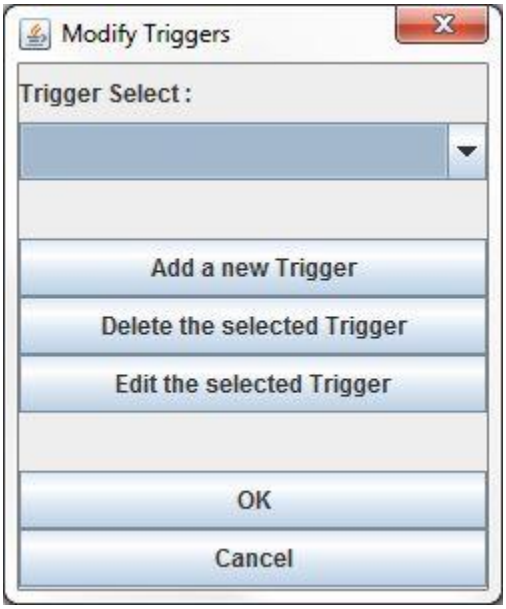
Next from the **Options** menu click on the **Baudot & FSK Options item**. Now select the options "75 Baud" and "850 Hz" as you can see below ..



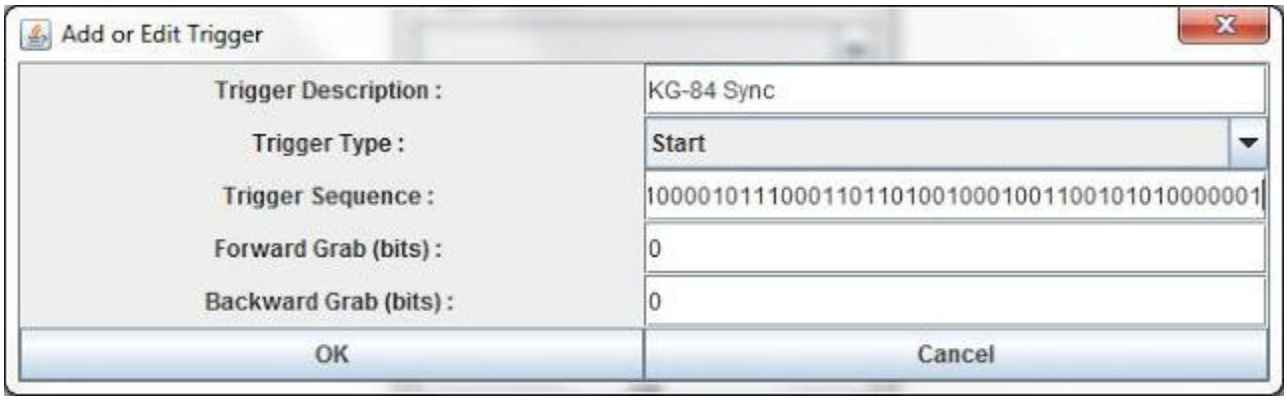
(Ignore the Stop Bits section which doesn't apply here). Now if you were to load the KG-84 sound sample all you would see is a lot of a binary data. Now it is common knowledge that KG-84 crypto systems use the following binary sequence for synchronisation ..

1111101111001110101100001011100011011010010001001100101010000001

So what we shall do is make this sequence so it is a Rivet start trigger. To do this click on the Triggers menu followed by the "Add , Edit or Delete a Trigger" item. Once you do that then this dialog box will appear:



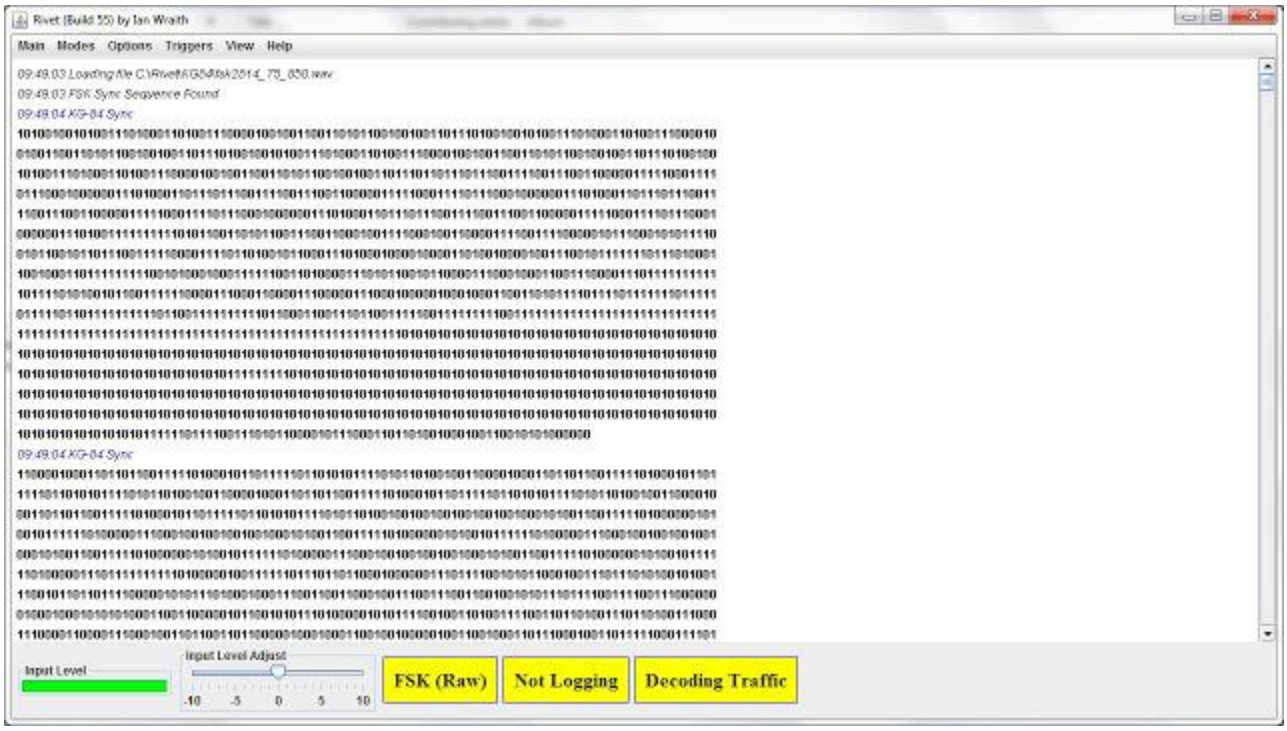
Now click on the "Add a new Trigger" button at which point you will see another dialog box which you need to fill in to look like this:



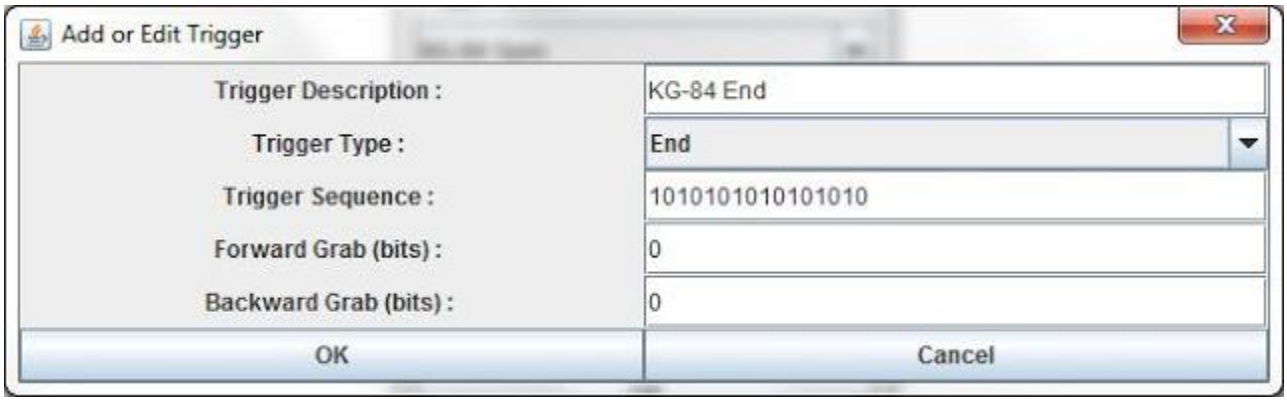
(you can cut and paste the binary value from this blog post rather than typing it in yourself). After that click on the OK button in this dialog box followed by the OK button in the Modify Triggers dialog box. Once you have done that the program will return to the main screen and automatically save the trigger to your hard disk.

Next you need to enable the new Trigger. Click on the Triggers menu and you will see the name of your new trigger with a radio button next to it. Click on this to enable the trigger as you see below.

With that all done either tune into a KG-84 transmission or load the recording I mentioned earlier. When you do that you will see something like the picture below.

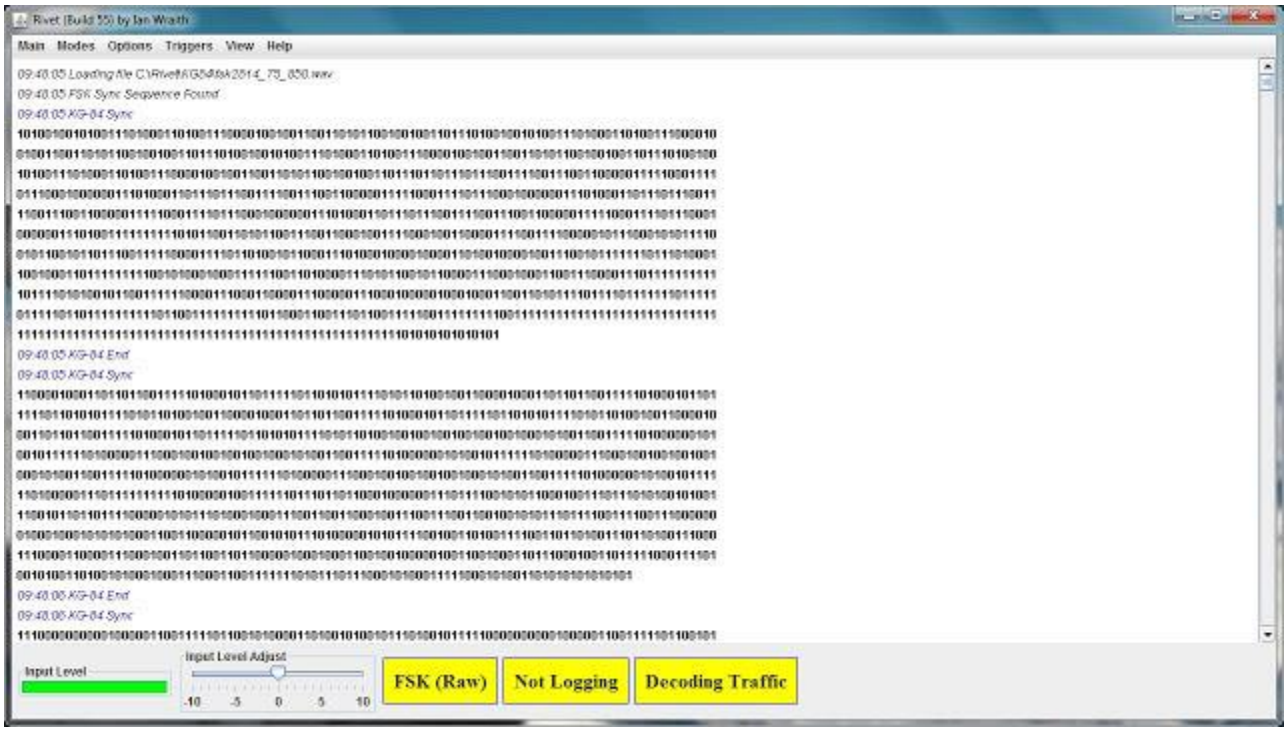


You can see from the words "KG-84 Sync" in blue (this triggers name) that the trigger activated several times. The display shows the trigger sequence followed by the data transmitted afterwards. Now that's all well but the display is cluttered with the alternating bits that are transmitted between the messages. It would be better get rid of those. We can do this by setting the alternating sequence of bits ***1010101010101010*** as an end trigger. So when Rivet detects this sequence it stops displaying decoded data until another start trigger sequence is received. To add this trigger click on the Trigger menu then again on the "Add , Edit or Delete a Trigger" item , then on the "Add a New Trigger Button" and fill in the dialog box that appears in the following way.



Click once again on both OK buttons and then enable the new trigger.

Now if you try to decode a KG-84 recording or live transmission you will see something looking like this …



Both types of triggers can be seen (in blue text) when activated and the KG-84 messages (or blocks of a message) are clearly defined.