Windows Software

For Windows, there are two programs, one for sending and another for decoding.

Executable Download

These programs use only the most basic functions of the Windows API so they should work on any flavour of Windows.

> Sender program for Windows: ebnaut-tx.exe
> Decoder program for Windows 32 bit: ebnaut-rx.exe
> Decoder program for Windows 64 bit: ebnaut-rx.exe

Source Download

If you want to modify the programs or see how they work, download the source below:

> Sender program for Windows: ebnaut-tx.c
> Decoder program for Windows: ebnaut-rx.c

The program expects to be compiled with the MinGW compiler, either natively on Windows or cross-compiled on Linux. The source make use of GNU 99 extensions so you must use the compiler option `-std=gnu99`.

Sending

The sender program drives a relay via the DTR and RTS modem control lines of an RS232 port. You must provide a GPS or Rubidium stabilised carrier and arrange a relay (or perhaps a balanced modulator) so that it can reverse the phase of the signal. This can be done with a small audio transformer and a double pole double throw relay. If one of the windings of the transformer has a center tap you can use a single pole double throw relay.

The RS232 modem control lines can only drive a very small amount of current, just a few mA. This may be enough to drive a small DIL relay. You might need some diodes to get the relay to toggle properly from the bipolar DTR and RTS signals.

There are two output options:

- **DTR + RTS**. DTR and RTS toggle together with the same polarity so that you can parallel them to obtain twice as much current.
- **DTR - RTS**. DTR and RTS toggle with opposite polarity so that you can make use of twice the driving voltage.

The RS232 voltages are typically +/-12V on a desktop PC and +/-5V on a laptop PC or USB serial device.

On a 9 pin D-type

- DTR is pin 4;
- RTS is pin 7;
- Ground is pin 5;

For anything other than a very small relay you will need to rig up a simple driver stage.

The relay arrangement produces hard switching of the carrier which creates a wide spectrum of keying clicks. This is not a problem at VLF but if you are attempting to operate in an LF band it is strongly recommended to use some sort of balanced modulator to produce the phase reversals. Then you can slightly smooth the DTR/RTS switching signal to soften the edges before using it to modulate the carrier.

The sender has a test function to exercise the modulation relay. Put in a symbol period of say, one second, and press 'Test'. This toggles the relay on and off at the symbol rate so that you can check that it is switching reliably.

Decoding

The decoder program demodulates the signal and decodes the message. It expects the received audio to be presented in a WAV file. You must use Spectrum Lab to receive and timestamp the signal. Spectrum Lab will filter the signal and apply strong sferic blanking, and take care of timestamping and sample rate drift correction. It will shift the received signal to baseband and output a WAV file containing the I/Q channels which the decoder requires. The header of the file contains information about the start time and sample rate of the recording and the decoder relies on this to work out where to find each symbol in the recording.

It is usually advisable to set Spectrum Lab to begin the recording a few seconds before the pre-arranged start time of the message, and to continue recording for one or two extra symbol periods at the end.

The decoder is very simple to operate. Select the code scheme, symbol period, and number of characters to match what the sender is using. Set the time offset to the amount of pre-trigger you have used in the Spectrum Lab recording.

The decoder will display anything it finds. It continues searching even after a successful decode and will update the display if a stronger decode (fewer symbol errors) is found at a better reference phase.

An output file is produced which has the same name as the input file but with `.wav` replaced by `-log.txt`. This is a plain text file containing progress and diagnostic messages and information about each decoded message.

The decoder will continue running for a long time as it tries wider and more varied reference phase patterns.

The decoder program can be launched from the command line. Running `ebnaut-rx -?` will output a summary of the command line options:

```
-N nchars    Message length in characters
-f filename  Input file name
-c ncpu      Number of CPU cores to use (default 1)
-L size      Viterbi list size (default 20000)
-p poly      Polynomial set
-F freq      Frequency offset, Hz
-T secs      Start time offset, seconds
-S secs      Symbol period (default 1.0)
-PS15        Phase search step 15 degrees
-PS30        Phase search step 30 degrees (default)
-a           Start automatically (default start button)
```

To the polynomial option `-p` you can give the polynomial set in comma separated octal (each polynomial must have leading zero), or use the short alias. For example, the following two options are equivalent:

```
-p 4K14A
-p 021113,023175,035527,035537
```

See Tables for the list of available polynomials. You can invent your own polynomial sets, you are not limited to the ones built into the programs.