

Software for Contesters – Jan/Feb 2008

### Contest Simulator Update

Early in the life of this column, I reviewed CW and contest simulator software for Windows XP. Since then, not surprisingly, there's been some progress, and one old favorite has found new life.

In that early column, I bemoaned the fact that the venerable DOS CW simulator, *Rufz*, was not available for XP. Ask and you shall receive... thanks to DL4WW and IV3XYM, *RufzXP* came out in 2006, and the latest (Version 1.1.2) is now available at [www.rufz.net](http://www.rufz.net).

For users of the DOS version, the way *RufzXP* works will be pretty familiar, but the *Windows* user interface (Fig. 1) is a big improvement. One unusual refinement, which I think was also available in the DOS version, is a Visually Impaired Graphic Mode, which displays the users' score in a format readily decipherable by screen reading programs.

Current score	Wrong callsigns	Main initial speed	Max speed	Min speed
0	0	CPM: 151	CPM: 0	CPM: 0

**RufzXP - Tancredi**

Current speed	Max points	Gained points	Errors	Time needed
CPM: 151	0	0	0	0 ms

Enter your call:

Volume: 50

Even if you're not particularly interested in setting high speed CW records, *RufzXP* is very useful in helping you to refresh your familiarity with real callsigns before the start of the contest season. You can use custom callsign files, and *MEdit* from VE3NEA (<http://www.netvampire.com/ham/>) will convert Master.dta files from K5ZD to the simple ASCII format that *Rufz* requires. You can also use this capability to convert specific files – for example, master files containing only DX or only US/VE calls. I personally find *Rufz* a little depressing, watching my accuracy disintegrate about 50 WPM, but it points out which letters and numbers I need to be careful about, even at lower speeds.

Meanwhile, VE3NEA's *Morse Runner* continues to improve since its last mention here, and has pretty well taken over the pileup simulator field, particularly since it's freeware. The latest version, 1.68, can be downloaded from <http://www.dxatlas.com/MorseRunner/>, and enables you to simulate the operating keystrokes required by *N1MM Logger*, *TR Log*, *Writelog*, *CT* and *NA*. It uses master.dta files to supply callsigns, so you can use K5ZD's files in their original form. Its capabilities for QRM, QSB, noise and flutter make it about the best simulator I've used.

It was only a matter of time, and AC7FA has now come up with a way to use *Morse Runner* with *N1MM Logger* to produce a true N1MM simulator. He did this by writing a script for use under a very powerful freeware keyboard mapping program called *AutoHotkey* (more on this below). You can download the script from [http://ac7fa.net/?page\\_id=11](http://ac7fa.net/?page_id=11). Both a stand-alone version and one requiring *AutoHotkey* on your computer are available. The simulator only supports WPXCW (with serial numbers, which is a lot better practice than CQWW, for example), but it makes use of all of *N1MM Logger*'s features, including ESM (Enter Sends Messages) mode. Very neat for working the kinks out before the next contest. Details on Don's web site and on the *N1MM Logger* web-site ([www.N1MMLogger.com](http://www.N1MMLogger.com)).

For several years now, I've been recommending *AutoHotkey* (<http://www.autohotkey.com/>) to people who want to remap their keyboards and automate frequently-performed tasks. To be honest, it is probably overkill for those tasks, but for something tricky like AC7FA's application, it's terrific. I use it for much more mundane things, like mapping a bunch of CW function keys and common multi-key N1MM Logger hotkeys to the numeric keypad on my computer. Laptop users and users of computers with non-English keyboards will find it useful for remapping needed keys to easier-to-find keys on their machines. When not in use, it is totally unobtrusive, and when you need a script, you just load it into the System Tray with a double-click. Just don't forget it's there, because unless the script writer has been careful to specify which windows are involved it can hijack keystrokes that you don't want hijacked. Don't ask me how I know!

### Schematics and Block Diagrams

From time to time on one of the Internet reflectors, someone we admire as a serious station-builder will say something like, "Of course, you keep all your schematics and block diagrams in a notebook, don't you?" I'm embarrassed to say I'm nowhere close to this ideal, at least in part because I don't like the looks of my hand-drawn diagrams and I'm too cheap to buy a copy of Visio, which seems to be the de facto standard.

In 2004, I'd been looking around for an inexpensive schematic drawing program for some time when I came across *ExpressSCH*, freeware from <http://www.expresspcb.com/ExpressPCBHtml/Download.htm>. ExpressPCB's business is making custom and production 2- and 4-layer printed circuit boards, and features very fast turn-around times (1 or 2 days, typically, on small orders). But even if you never lay out or buy a printed circuit board from ExpressPCB, their schematic-drawing software is terrific. The user interface is readily learnable, and flexible enough so that you can draw circuit diagrams, block diagrams, and just about anything else small-scale. In this respect, it is much like *TinyCAD*, mentioned in a recent QST, but I prefer *ExpressSCH* because for schematics, it uses US-standard symbols instead of the new (and still unfamiliar to Americans) European standard.

There's no such thing as a free lunch. Like every schematic drawing program I've seen so far, *ExpressSCH* is tailored to the needs of IC circuit designers. It has a huge library of digital and analog ICs, but only a few RF-related components and schematic symbols. Plan on spending some time with the program's component and symbol creation capability reproducing the standard table found at the front of every ARRL *Handbook*. Fortunately, this is easy to do; for example, creating a 3PDT relay from the SPDT relay already in the library took only a minute or two, and once you've done it you can store the custom symbol for retrieval whenever you need it again. Creating tubes from scratch is a little tricky (one hint is to work with the symbol in a diagram that already has other components, so that you can get the size right), but each only needs to be done once.

As I mentioned, I now draw station block diagrams with *ExpressSCH* as well – it's not intended for this, but the simple drawing tools provided give you all the capabilities you should need for this. Figure 2 is a good example.

