

## Digital Contesting

### *TinyFSK*

RTTY contesters who use FSK may want to consider the *TinyFSK* method which offloads the Baudot bit timing to a dedicated Arduino processor, independent of the Windows environment. Andy Flowers, K0SM, wrote the *TinyFSK* program in 2013. It runs on the tiny Arduino Nano processor with the addition of two NPN buffer transistors between the Nano's output pins and the radio's FSK and PTT inputs.

At the time, *TinyFSK* users had to homebrew this hardware and cable that connects from a PC USB port to the radio FSK and PTT inputs. Figure 1 shows the first *TinyFSK* I constructed. The two NPN transistors with their base resistors are point-to-point soldered right on the Nano board, as is the two-wire-plus-shield output cable to the radio. A Micro-A USB cable connects between the PC and the USB connector on the Nano. The plexiglass Nano case was found on eBay from a German supplier. This particular unit has a DE15 connector on the radio cable for connection to the Elecraft K3 (and now K4) Acc jack, where the FSK and PTT inputs reside.

### *Mortty* Version 1

In 2018, I sent one of my *TinyFSK* units to Larry Gauthier, K8UT, in

conjunction with our discussion about *TinyFSK* implementations. Larry extended my design to use a PC board for the Nano to plug into that also contained the two NPN transistors and resistors plus a 3.5-millimeter TRS jack for the radio cable. He and Steve, N8AR, worked together on the new design and packaged it in a 1 × 1 × 2-inch aluminum case.

They produced a construction and programming manual with parts list for DIYers to build their own hardware platform for the *TinyFSK*. The manual was offered free, but demand grew for them to bulk-buy the parts and sell kits. Soon after offering the kits on April 2, 2018, they discovered that the K3NG CW Keyer program would also run on their unit with a couple modifications to the PC board and parts list.

### *Mortty* Version 2

On June 13, 2018, *Mortty* was introduced as a kit that could be programmed to run either the *TinyFSK* or K3NG CW Keyer “sketch” (Arduino-speak for “program”). Two miniature LEDs were added to show PTT and CW/FSK.

Although the initial kit for *TinyFSK* was unnamed, it was retroactively called *Mortty* Version 1 and this new kit for either sketch was dubbed Version 2. Even more retroactively, I now

refer to my original homebrew project as “*Mortty* Version 0”

### *Mortty* Version 3

On July 4, 2018, *Mortty* Version 3 was introduced with a number of nice improvements. Notably, some PCB pads were added to allow programming with a third sketch called the nanoIO written by David, W1HKJ, that works in conjunction with *fldigi* software to dynamically switch between the *TinyFSK* and K3NG CW Keyer sketches. This eliminates the need to load a new sketch when switching between CW and FSK.

In addition, the 3.5-millimeter output phone jack was upgraded to include an SPST switch that operates based on whether a plug is inserted into the jack. This was used to switch the Nano from program mode to operate mode when the plug is inserted.

*Mortty* Version 2 had a jumper inside for the same purpose but now the user didn't have to open the case each time a new sketch was desired. The need for switching between program and operate mode was discovered in some scenarios such as running the K3NG CW Keyer with *N1MM Logger+*.

One of the most frequent customer requests was a speed pot for use with the K3NG CW Keyer. On May 14, 2019, Steve and Larry introduced the *Mortty Speed Pot* option which could be retrofitted to *Mortty* Versions 2 and 3. Not to be outdone on cramming lots of capability into a small package, one enterprising DIYer shoehorned four memory pushbuttons and a speaker into a *Mortty* along with the speed pot.

### *Mortty* Version 4

*Mortty* Version 4 was introduced on July 17, 2020, as preassembled-only, using SMD components, two additional LEDs (power and 2nd CW for SO2R/nanoIO FSK), speed pot included and a TRRS paddle jack



W0YK homebrew hardware for *TinyFSK* (“*Mortty* Version 0”).



**Mortty Versions 1, 2, 3, and 4.**

providing a second ring connection for DIYers to add options. Version 4 details are on the *Mortty* website, <https://hamprojects.info/mortty/>.

### Sketch Swapping

Although sketches are easy to load into the Arduino, I much prefer to dedicate each hardware unit to one application and just have an abun-

dance of units to cover all my CW and FSK needs in different locations and for traveling. They are so small and inexpensive, I just consider them as a “lumpy” cable embedded in my various stations. Accordingly, I have several *TinyFSK* sketches loaded into a number of Version 0, 1, and 2 units (without speed pots) and another batch of Versions 2, 3, and 4 with

speed pots for CW. Figure 2 shows Versions 1 through 4.

### Summary

*Mortty* is a very compact and reliable solution for many CW and FSK applications. Thanks to KØSM, N8AR, K8UT, and W1HKJ for their innovative contributions.

## The Little Pistol Pages (Continued from page 26)

QSO B4” responses that eat up your S&P rates. You need no paper on your desk, and you can discard the pencil.

The computer can send all the CW you require, if that’s your preference. You don’t need a key at hand. It throws out your call sign or a CQ, gives all the responses, and says 73 and the ham equivalent of “who’s next?” (QRZ?). If you made an error in your call sign copy, the computer will note the typed correction and fix it in your log file as well as in your last transmission to the station. The computer will total your score, keep track of your ongoing rate, show you multipliers you still need, and insert repetitive log data, so you don’t have to endlessly write “599.”

At the end of the contest, the computer generates a summary of your activity, and puts out the proper Cabrillo-format file, that you can then

email to the contest sponsors. All your post-contest paperwork, done in a half hour!

The contest logging software used almost universally is *N1MM Logger+*. It’s free, well documented, and inclusive of almost every amateur radio contest in existence. Most of your friends are probably using it already, and lots of experts are available to help when you run into trouble.

### Run, Run, Run!

For efficient S&P, the band must be full of stations that you have not yet worked. When you find your S&P rate has dropped below an acceptable level, it’s time to change bands or to start running yourself.

That said, to keep up a good average rate over the course of a contest, you must run and do it extensively. Otherwise, you are only

sampling half of your possible QSO market. The other half, the other S&P ops, you will never work. So, for Little Pistols, the idea is to S&P until things get slow, and then start running. But, don’t wait until the bands are dying.

Running requires lots of practice. You may attract pile-ups, and then you must try to pick out at least one responding call sign. To reduce the pile, copy part of a call sign so you can respond with “K4B?” or “VE3?” Work your way through the pile-up that way.

Skills you will need include the ability to copy call signs accurately and type them into the correct field in the computer logging screen. Practice these skills daily, until you feel your abilities are adequate. The faster you operate, the higher your rate. In contesting, rate is everything that drives us.