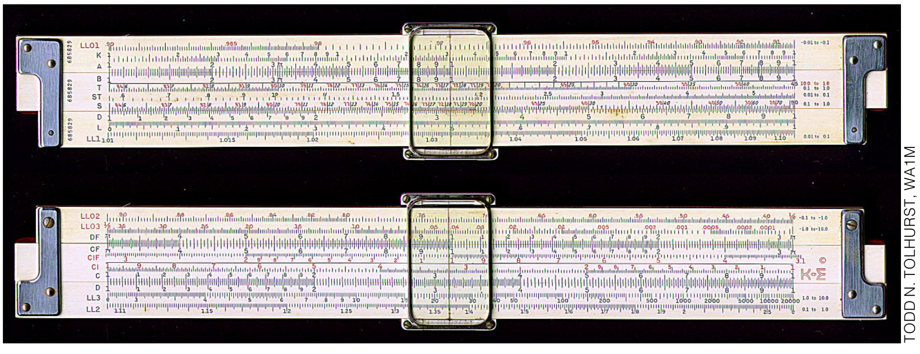
By Brian Wood, WØDZ

The Return of the Slide Rule Dial

Transceivers have long abandoned the elegant convenience of analog slide rule dials. But if you own a Yaesu FT-1000 transceiver, the slide rule dial can live again on your PC. Readers familiar with *Visual BASIC* can also adapt this clever software for other computer-compatible radios.

There's a line in a classic *Star Trek* episode that goes "Here you stand—the perfect symbol of our technical society—mechanized, electronicized, and not very human." The line was spoken by Karidian, once known as the evil "Kodos the Executioner," about Captain Kirk. It could just as easily have been about modern amateur radios. In the "good old days," when Amateur Radio was younger, radios had an ambience (heat, light and smell) that doesn't exist today in our "microprocessorized" versions. (Of course, they also had drift and accuracy problems, contact arcing due to the high voltages, and a lack of features we accept as essential today, but I digress.) I eagerly anticipated the digital revolution that was coming as much as anyone, but I now look back and see that something was lost in the transition–the *information* that was available in the display. This article looks back fondly at the old rigs and offers a modern technological solution for at least some of that missing ambience in the form of a *Visual Basic* program that gives the Yaesu FT-1000 (and potentially other modern radios with RS-232 interfaces) a "slide rule" dial.

"What's a slide rule?" you might ask! If you hit grade school after about 1973, you might never have experienced one. Nestled in history between the abacus and the handheld calculator, they were flat, plastic sticks that slid against each other, with numbers printed on them, and they had a clear, plastic, movable dial with a thin line down the center. They allowed engineers and scientists to make calculations. If you saw the movie *Apollo 13*, you saw NASA engineers using them as



TODD N. TOLHURST, WA1M

Figure 1—Technological antiques: the venerable slide rules.

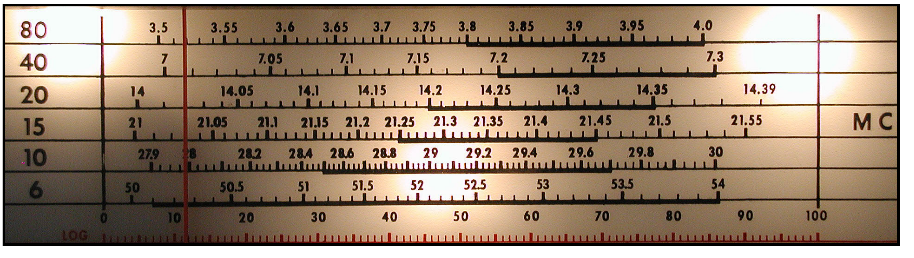


Figure 2—The slide rule dial made famous on the National 270.

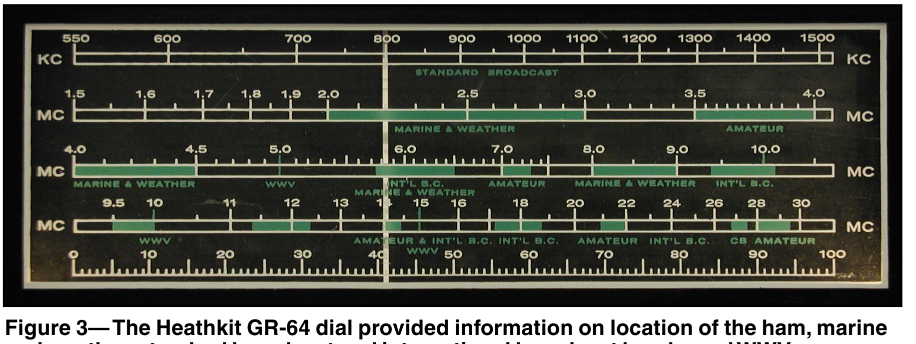


Figure 3—The Heathkit GR-64 dial provided information on location of the ham, marine and weather, standard broadcast and international broadcast bands, and WWV.

men traveled to the moon. How crude, you say! But they worked, they were accurate to about 2 decimal places, and they didn't require batteries. See Figure 1.

Actual vs Displayed Frequency

The frequency tuning mechanism in older radios consisted of a variable capacitor or inductor that was part of the LC circuits in a VFO. When you turned the tuning knob, you weren't putting pulses into a microprocessor that was computing the frequency, outputting data to other circuitry and updating a digital display. You were actually directly varying an oscillator's frequency. The capacitance of a variable capacitor as it is tuned across its range, while predictable, is not linear and also varies slightly from batch to batch, so it was difficult to make a display that accurately represented the rig's frequency. Lots of creative solutions arose, and one of them was the slide rule dial.

The slide rule dial consisted of a vertical "dial pointer" that moved across a long, translucent background that had frequencies printed on it and light bulbs behind. A dial cord was strung around the main tuning capacitor and connected by pulleys to the dial pointer. Great effort was expended to make the frequencies printed on the dial face match the actual frequencies that were being generated or received. Since production variations had to be accounted for, methods of calibrating the dial had to be made available to the user. Thus was born the "100-Kc crystal calibrator," a circuit that injected carriers every 100 kHz across the bands when it was turned on. A tweak to the position of the dial pointer was available to allow the dial to be adjusted at a 100 kHz mark near the operating frequency. Figure 2 shows a National NC-270 dial that was built in this way.

You are Here

Although the slide rule dial was born of necessity, one of its byproducts was the ability to mark the bands with information. As you can see from the NC-270's dial, the locations of phone bands prior to 1968 was visible as thicker lines. The venerable Heathkit GR-64 shortwave receiver (Figure 3) provided information on location of the ham, marine and weather, standard broadcast and international broadcast bands, and WWV. Of course incentive licensing changed the location of the accessible phone bands for some hams, and other rules changes have affected the starting points of the phone bands themselves.

Once radios went "digital", that is, started displaying operating frequency as

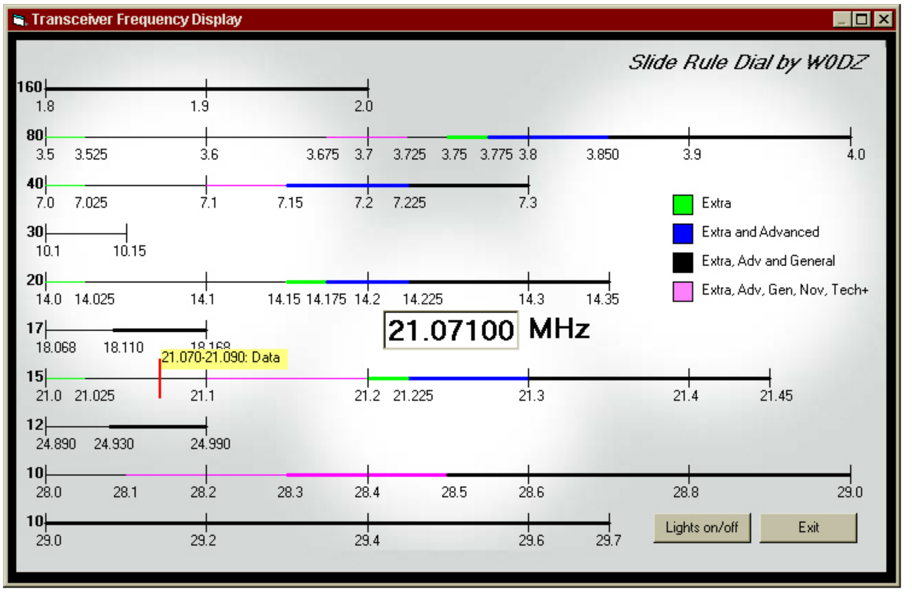


Figure 4—Here's my *Visual BASIC* implementation of a modern "slide rule dial."

```
Private Sub Timer1_Timer()
Dim x() As Byte
Dim Freq As Double
Dim Counter As Integer

    ' Send "Get current operating data" command to FT-1000
    MSComm1.Output = Chr$(0) + Chr$(0) + Chr$(0) + Chr$(2) + Chr$(16)

    ' Wait for buffer full
    ' Note: this is highly dependent on processor speed and ought to be changed,
    ' but it's an easy way to avoid hang-ups.
    Counter = 0
    Do
        Counter = Counter + 1
    Loop Until MSComm1.InBufferCount >= 16 Or Counter > 20000

    ' If no data were read, skip the rest of the routine!
    If Counter >= 20000 Then
        GoTo Skip
    End If

    ' Read data from rig
    x = MSComm1.Input

    ' Calc new freq
    Freq = x(1) * 256 + x(2)
    Freq = Freq * 256 + x(3)
    Freq = Freq * 256 + x(4)
    Freq = Freq / 16
    Freq = Freq / 10 ^ 5
    Text1.Text = Freq

    ' Rest of code to position red dial indicator, pop up voluntary band plan,
    ' and position memory markers goes here.

Skip:
End Sub
```

Figure 5—Code snippet showing how to extract the frequency, band and mode data from the FT-1000 and show the frequency in the text box in the center of the display. The RS-232 interface uses the mscomm32.ocx control, called "MSComm1."

digits, either with Nixie tubes, seven-segment displays or alphanumeric displays, this wealth of information—your *place* in the spectrum—was lost. Today, even the most expensive rig does not show you the edges of the ham bands or sub-bands, or the locations of international broadcast bands. It behooves you to keep a chart handy so you don't inadvertently stumble into the wrong section of a band. The 30, 17 and 12-meter bands are the worst. I can never remember exactly where they are!

Why is this? How did we manage to add so many cool features to our radios, but lose one of the most important? I think you don't have to look too far to see that technology for technology's sake is rampant today. Electronics has matured to the point that we can now make almost anything we want, and we do—without paying attention to the human beings who have to operate them. It's time to restore some of the "humanity" to our technological marvels and make things that do our bidding, not that make us think like a machine. Paradoxically, technology itself can provide the answer.

Technology to the Rescue!

There are several good control programs for PCs and Macs that are available for modern ham rigs. All of them, though, mimic the look of the radio and let you work the controls from the computer. I don't know about you, but I *like* operating my rig—pushing buttons, turning knobs, flipping switches. That's more fun than clicking a mouse. What I want from my computer is to put it to work giving me the information that's missing from my radio. I just want a new display!

With that in mind, I thought it would be fun to write a *Visual BASIC* program that could recreate the ambience of an old radio with a slide rule dial, but that could take advantage of today's technology. The screen shot shown in Figure 4 is the result. The ham bands from 160 through 10 meters are shown, and the various sub-bands for each class of license are also shown. Even the voluntary band plans are shown; as you pass through the relevant section of a band, a popup text box explains the details (one example of which is shown in the screen shot). As with the NC-270, the phone bands have a thicker line marking their location, but they're color coded to show the incentive licensing sub-bands. Unlike the radios of old, modern radios don't have a logarithmic effect; the displayed frequency changes at a constant rate as you move the dial. I even threw in a "backlit" display—a background image (jpeg), created using Adobe *Photoshop*, which mimics three light bulbs behind the display. There's a button at the bottom to turn the "bulbs" on and off (and these bulbs will never burn out!). Since the band edges are likely to change again in the future, the program is written to allow reasonable modifications, although they aren't necessarily trivial.

The program's dial pointer is a red, vertical line that moves across the operating band. On old radios, when you changed bands the pointer stayed put, so you ended up on whatever frequency the pointer happened to be pointing to—sort of a "mechanical memory." With electronic memories, this is not true today, so the dial pointer does not need to run the full height of the display. As you change bands, a marker is deposited at the last location of the pointer and the pointer moves to the memorized location on the new band. Since the program was written for a rig that has two memories for each band, two markers, one blue and one green, can be deposited on each band. When you return to a band, the relevant marker is replaced with the moving red dial pointer.

Program Details

I wrote this program to work with a Yaesu FT-1000MP, but it could be adapted to virtually any modern rig. As shown in the code snippet in Figure 5, it doesn't take much code to output a command to the rig that asks, "What band, mode and frequency are you on?" and then get the reply. Note that this code is placed inside a timer event that is set to occur every 100 ms, so that the display can keep up with the user.

The FT-1000MP returns binary data that must be converted into a useful number, but once you have the band, mode and frequency from any rig, the rest of the program would remain essentially the same. (By the way, I discovered while writing this program that the FT-1000MP manual, at least mine, dating back to 1999, is wrong—the frequency is returned as a pure binary number in several bytes, not binary-coded-decimal as the manual shows. It took some experimenting to figure out what was really coming back from the rig!

I also discovered that there's no way to read the front/rear band information, which means the program must start without any markers on the screen; you have to go through the bands once to "teach" the *Visual BASIC* program where the memories are. For the same reason, you also have to stick to a strict discipline of using a different mode for the front and rear memory so the program can tell that you've pushed the button!)

Visual BASIC programs are hard to print because of the large number of graphical properties that are attached to each object, so it is impractical to print the program in this article. However, I'll be happy to provide the source code and/or the executable free by e-mail (**w0dz@arrl.net**). If you don't have e-mail, send a floppy disk and an SASE to the address shown below.

As a final note, I'd like to implore ham radio manufacturers to think more about the displays they put in the next generation of radios. Many new radios have larger displays with spectral information, but they still don't show you band edges or other information. Let's see some more information content in the displays!

You can contact the author at 710 Grove Ct, Loveland, CO 80537; **w0dz@aarl.net**.