

ERC-M Rotator Controller and *PSTRotatorAZ* Rotator Control Software

Reviewed by Pete Smith, N4ZR
NCJ Contributing Editor
n4zr@contesting.com

In this era of software defined radios and surface mount components, one of the remaining pleasures for those of us with an urge to build things is in the area of station control. I've done quite a bit of station automation, but sometimes, with two rotators, I still feel a little like a one-armed paper hanger. I went looking for a relatively simple, low cost way to automate control of my rotators, while not adding anything to my operating desk if I could help it.

A Solution from Germany

Casting around through all the usual sources, one of the first Google hits was the Easy Rotor Controller (ERC) from the German firm Schmidt-Alba. Schmidt-Alba's proprietor, Rene Schmidt, DF9GR, is a well-known member of the Bavarian Contest Club. His ERC controller line includes one- and two-axis rotator controllers, assembled or in kit form, that can connect to virtually every rotator ever made. The controller does the heavy lifting in firmware, translating commands from your logging or station control software, as well as storing calibration, rotation stops, and so on. It connects to control-ready rotators through a six-pin mini-DIN connector.

For those rotators that aren't control-ready, Rene offers a couple of solutions. One is a "Rotorcard" that mounts inside the existing controller, providing relays that emulate pressing the left or right rotator controls. Another is the ERC Version 4 kit, just announced. It has the necessary relays included and is designed to mount inside most rotator control boxes. For one rotator,

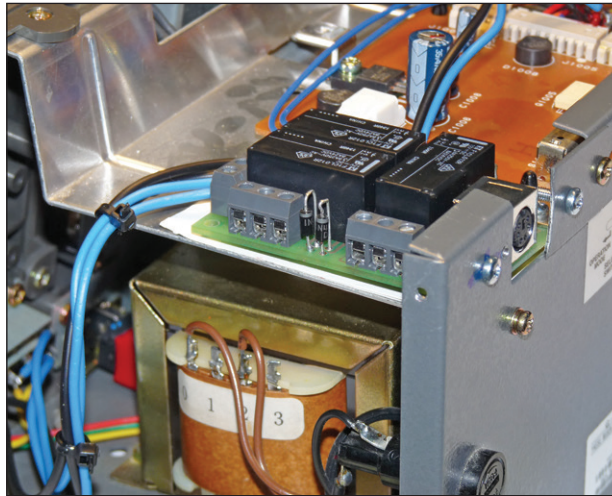


Figure 7 — The completed Rotorcard installed in a Yaesu G-800SA rotator control box.

Version 4 would be the easiest, cheapest solution by far.

ERC-M is the two-axis model, which can be used either for azimuth-elevation (az-el) control in a satellite or EME system, for example, or to control either one or two azimuth-only rotators. All of Rene's controllers connect to the station computer through USB or RS-232 serial ports (choose the version you need when you order), while the ERC-M has a LAN option suitable for remote control applications. Finishing touches include either a desktop controller

Bottom Line

The ERC-M rotator controller, Rotorcard interface, and *PSTRotatorAZ* software provide tools to include control of one or more antenna rotators in a station automation system. Excellent vendor support is available to help with complicated requirements.

box with front panel displays and pushbuttons, or a minimal enclosure for behind-the-scenes control.

In my station, I needed control for two Yaesu rotators — one ready for computer control (a G-1000DXA) and the other in need of a Rotorcard (G-800SA). I wanted to rotate my tribander stack and a 40 meter short Yagi above the top tribander. The lower Yagi in the stack is on a side mount with less than 360 degree rotation, and the 40 meter Yagi is offset 90 degrees from the tribander below it, so I knew my requirements might be difficult.

A quick exchange of e-mails with Rene persuaded me both that his hardware would work for me and that tech support would not be a problem. So, off went my order and PayPal payment, about \$175 including shipping. I consider this quite competitive with other solutions available on the US market, particularly for two rotators. I could have added a desktop controller case, but I chose not to.

Getting the Hardware Ready

Only six days later, the package arrived. Inside were a ready-to-go ERC-M USB controller, a very small "slimline" enclosure, a Rotorcard in kit form, a couple of mini-DIN cables, and a CD with software and extensive, systematic documentation. The Rotorcard kit took me about an hour to build, soldering relays and a half-dozen through-hole components on a nice-quality glass-epoxy PC board. At first, I anticipated the hardest part might be installing the Rotorcard inside my G-800SA, but after a little thought and study of the instructions I felt confident to proceed. Six wires soldered



Figure 8 — The ERC-M rotator controller in its slimline case. A USB cable connects to the station computer, and both rotators are connected to a single 15-pin connector.

to readily accessible connections, guided by a schematic drawn from the original rotator manual, gave me all I needed, and the Rotorcard fit neatly into the controller cabinet (Figure 7).

For the other rotator, all that was required was to cable it to the ERC-M, with both cables passing through a single 15-pin D-SUB connector (Figure 8). Again, with care and a little thought, I had no trouble wiring the connector specifically to match the two rotators. You do need to be careful to label the cables, though, because the hookup is different for a Rotorcard than for the Yaesu rotator that is controller-ready, but both use a six-pin mini-DIN connector.

With this complete, I connected a USB cable to the ERC-M, and its driver self-installed on my *Windows 7* PC; the CD also included a driver for *Windows XP*, had I needed it. If you use a Rotorcard or have the RS-232 version, the ERC-M requires a 12 V dc supply (10 mA for the ERC-M, 300 mA for the Rotorcard). A wall cube power supply works fine. The USB version of the ERC-M without a Rotorcard can be USB-powered.

I then installed two pieces of software: a *Service Tool* that is used to test and calibrate the ERC-M, and a *Rotor-Control*, with two compass rose indicators, one for each axis. I should note that calibration was particu-

larly quick and easy, which I understand is exceptional in the rotator control field.

I tried the controller, and of course it didn't work. A couple of quick e-mail exchanges with Rene corrected my mistakes, and he also provided an update of the *Service Tool* to better manage the less-than-360-degree rotation of one tribander. Turnaround time was less than a day, due in part to a 6 hour time difference. As Rene observed wryly, "this rotator business can't be my only job," but in any case, both rotators were quickly dancing to my tune.

Integration with Logging Software

I then turned to the problem of interfacing the ERC-M to *N1MM Logger*, the logging and station control software I use during contests. This revealed an unexpected problem, because *N1MM Logger* requires one serial port per rotator, while ERC-M looks to handle all communications for both rotators through a single port. Rene had the answer, which brings me to the second part of the story. It would be necessary, he said, to use *PSTRotatorAZ* (*PSTRA*) by Codrut Buda, YO3DMU, between *N1MM Logger* and the ERC-M controller.

PSTRA costs 15 Euros by download, and I very quickly had a licensed copy of the software. There is a lot to *PSTRA* — one friend called it the "Swiss Army Knife" of rotator control software. It can act as a smart controller for virtually any rotator ever built, with lots of ways to tell the rotator(s) where to turn. It also functions as an intermediary between any of those rotators and a large number of logging programs. For my purposes, the essential thing it does is to read the UDP (User Datagram Protocol) messages that *N1MM Logger* sends to its own *N1MMrotor* program, which is not used in this case, and translate them into messages on a single COM port in the appropriate format to control two rotators through the ERC-M (Figure 9).

With some more long-distance help from Rene and Codrut, I soon had *PSTRA* controlling my rotators, and *N1MM Logger* talking to *PSTRA*. The one thing that was still lacking was proper handling of *N1MM Logger's* way of indicating the offset of my 40 meter Yagi. Until that was resolved, I could point my upper rotator to a station using the built-in N1MM command keystrokes, but on 40 meters it was not taking the offset into account. I mentioned this in passing in an e-mail to Codrut, saying

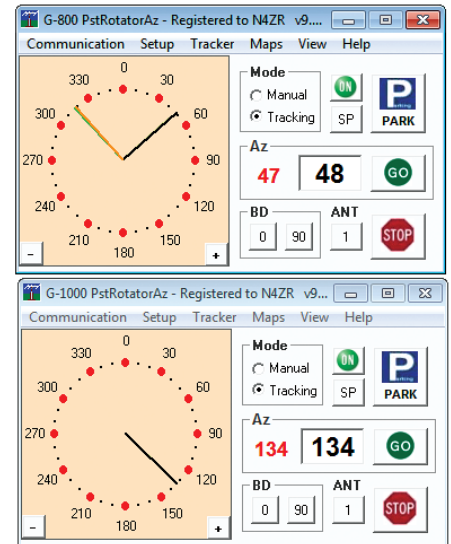


Figure 9 — The *PSTRotatorAZ* display showing the direction of both of the author's rotators.

I thought it was probably a matter for the N1MM team. The very next day, Codrut released an update of *PSTRA* that delivered a perfect solution.

A Happy Ending

Where do things stand now? With a stream of DX Cluster or Reverse Beacon Network spots displayed in *N1MM Logger*, I can click on a needed station and depending on the band, either the whole stack turns (for 20 – 10 meters), or the top rotator turns and points the offset 40 meter Yagi in the correct direction. It's all automatic. The ERC-M protects the lower tribander against over-rotation, and the calibration of both rotators is as good as I can resolve by eye.

Even more to the point, these two vendors have demonstrated that in the Internet era, distance is no longer a significant barrier to superb service and support. (I should add that neither Rene or Codrut knew that I was thinking of doing this review, so I believe the level of support is indicative of what any customer can expect.)

Manufacturer: ERC-M controller — Eida Alba de Schmidt, Kreuzangerstr. 58, 86399 Bobingen, Germany; <http://easy-rotor-control.com>. Price: ERC-M (USB version, assembled and tested), \$120. Rotorcard kit, \$23. Slimline case, \$22. (Prices are approximate, not including VAT or shipping. Pricing is in Euros, exchange rate will vary.) ***PSTRotatorAZ* software** — PstRotator. www.qsl.net/yo3dmu/index_Page346.htm. Price: about \$20 (pricing is in Euros, exchange rate will vary).