[Juergen E. Malner, NV1Q, photo]

Computer-Controlled Antenna Rotator

A simple Arduino-based control system.

Juergen E. Malner, NV1Q

My ham station is very automated. Clicking on a DX station that shows up on the cluster will set the frequency and mode on the radio, and pressing a button on the keypad will send my call in CW or activate the voice keyer. The only thing missing was the link between the logging software *DXLab* (**www.dxlabsuite.com**) and my Yaesu G-800DXA controller and rotator.

DXLab provides a serial port where the data for the short/long path is available, but the rotator lacked the needed serial board to process the data. I considered several options before settling

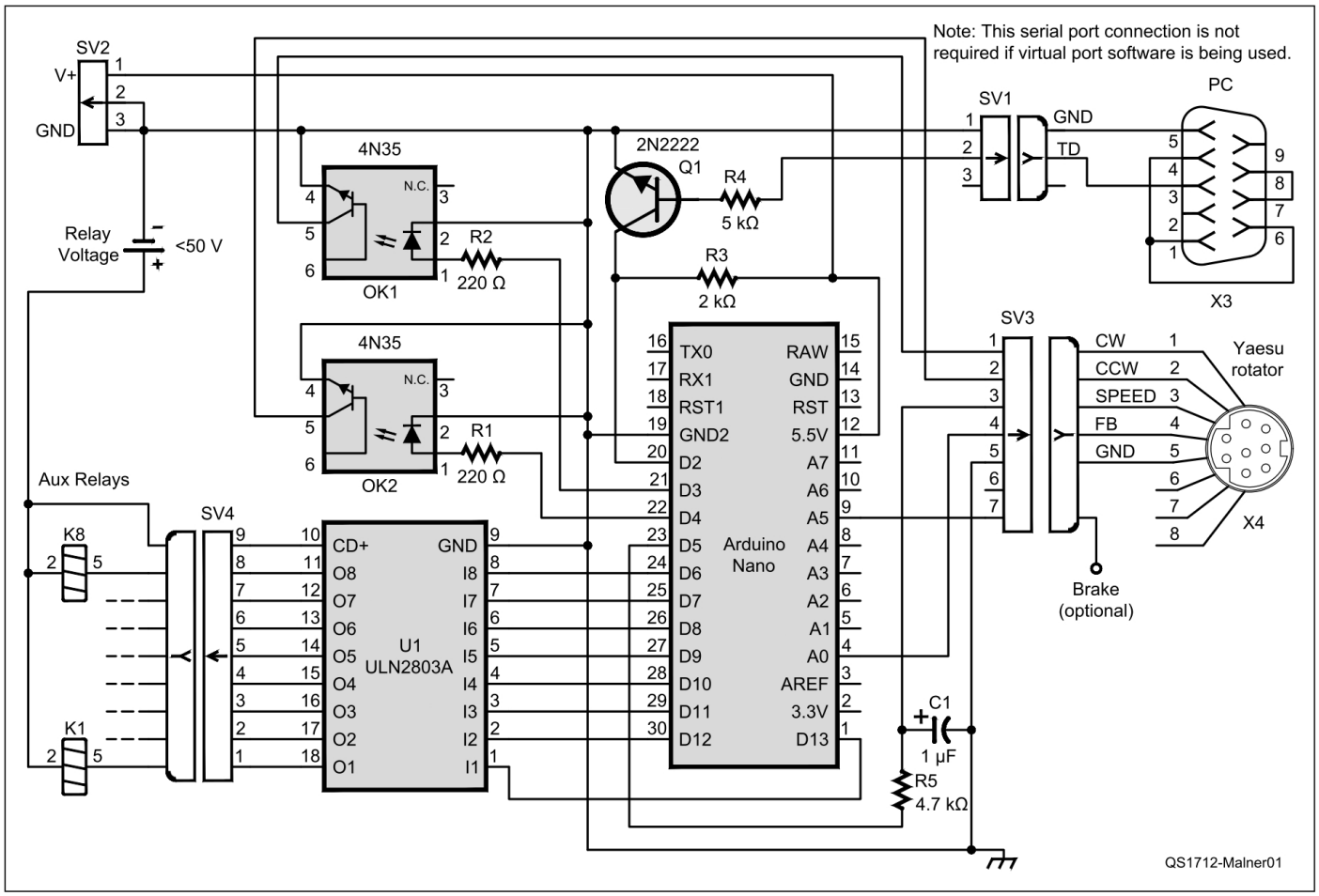

Figure 1 — Schematic of the complete system. Relays (contacts not shown) with relay voltage under 50 V can be used for switching antennas if desired.

on an Arduino computer-controlled antenna rotator, with azimuth direction displayed on the computer.

Hardware

The Arduino Nano is the main interface between the computer and the rotator controller (see Figure 1). Optocouplers OK1 and OK2 isolate the signals for clockwise (CW) and counter-clockwise (CCW) rotation. The rotator positioning feedback connects directly to an Arduino analog input. Transistor Q1 converts the 5 V Arduino output to a serial port signal on the computer. RC filter R5 and C1 smoothes the pulse width modulated (PWM) signal from the Arduino that controls the rotator speed.

You will need a ULN2803 Darlington driver if you use any of the available 5 V eight-channel relay modules. Optional relays K1 – K8 activate to switch antennas when the corresponding pin is pulled low.

Functional Description

A desired rotator position can be achieved in several ways. You can manually input commands using the

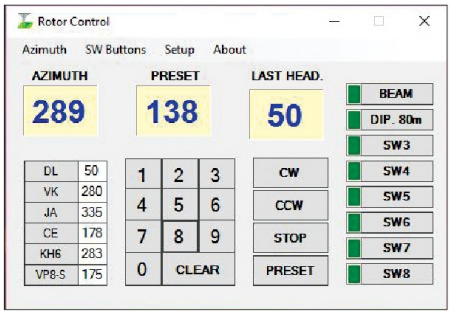

Figure 2 — The main rotator control screen shows buttons for up to eight auxiliary relays, and for pre-programmed countries. Relay buttons can be hidden if not used.

CW or CCW buttons (see Figure 2), directly input degrees using the display keypad, or use pre-programmed buttons. You can also left mouse click a cursor line on the azimuth map display (see Figure 3) to indicate the desired antenna direction. Finally, you can use your logging program (I use *DXLab*) by pressing either the short- or long-path button that indicates the antenna direction to a selected station.

When an azimuth direction has been selected, the program will send a corresponding number between 0 and 1,023 over the USB connection to the Arduino. This number is compared in the Arduino with the analog rotator feedback signal. The Arduino analog input accepts a voltage between 0 and 5 V, scaling to a value from 0 to 1,023. The Arduino then computes whether clockwise or counter-clockwise rotation will provide the shortest way towards the desired antenna direction, and energizes either the CW or CCW input at the rotator control unit. When the number from the PC matches the number from the feedback signal, the rotator is de-energized and rotation stops. The rotator speed will decrease prior to reaching the desired position.

Setting Up the System

The rotator ramp-down speed can be programmed in the rotator setup screen (see Figure 4), and you can enter and customize your call sign in the azimuth setup screen (see Figure 5). I use *DXLab*, so I have a hardware option and a software option from *DXLab* to process the antenna direction. In the hardware option, data is first sent to the Arduino Nano over the serial link and is then sent over the USB connection to the computer for further pro-

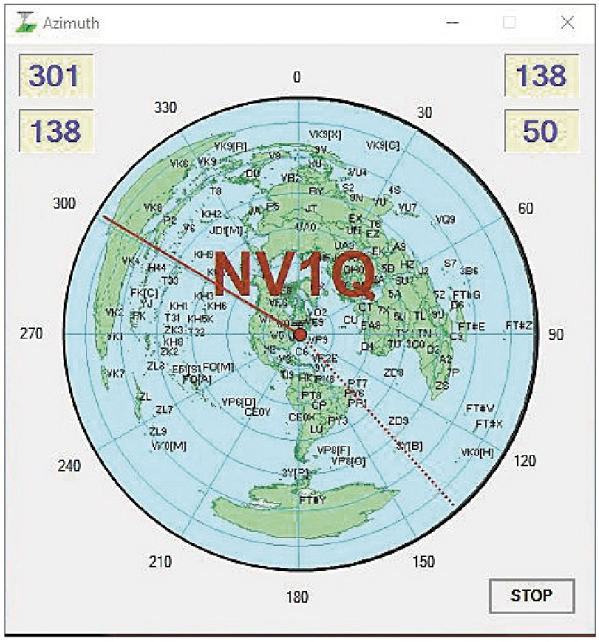

Figure 3 — Left-click your mouse to move the cursor line to the desired antenna direction on the azimuth map to initiate the rotator.

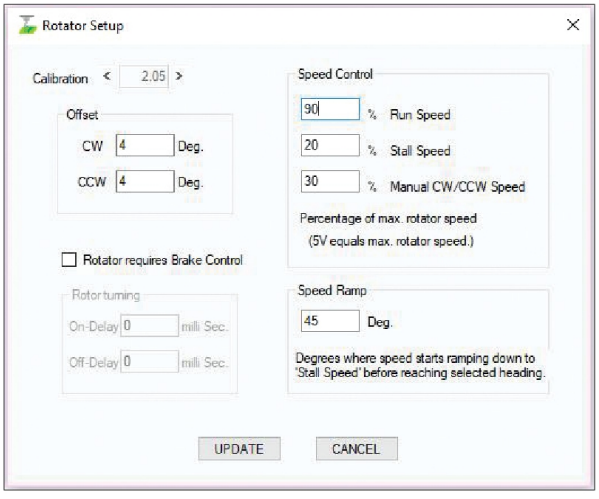

Figure 4 — Initial rotator setup.

cessing. In the software option, a virtual serial port can be installed. I use *VSPE* (*Virtual Serial Ports Emulator*, from **www.eterlogic.com**).

Setup options include selecting the correct serial ports for the Arduino and the virtual port software, labeling the relay buttons, assigning your favorite DX entity to a button, and calibrating the initial rotator direction.

I created my azimuth map using *DX Atlas* from Afreet Software (**www.dxatlas.com**), and centered it on my home grid square, FN32. You can import any JPEG graphic (instructions are provided in my software available on the **www.arrl.org/qst-in-depth** web page) and display it inside the circle.

I used Microsoft Visual Studio to write and compile the PC program, which can be installed with a single EXE file. The IDE to install the Arduino rotator software can be downloaded from the Arduino website (**https://www.arduino.cc/en/Main/Software**). During the installation of the PC software, a folder will be added on the desktop. This folder contains all the support documentation to set up and operate the NV1Q rotator system as well as the Arduino software.

The Yaesu rotator provides a mini DIN female connector where all necessary connections for operating the rotator with this design are available. A delay for energizing or de-energizing a brake, if required, can be set in the rotator setup screen (see Figure 4). The lead image shows my printed circuit board mounted inside the Yaesu controller.

Caution: Measure the maximum feedback voltage from your rotator. The Arduino can be damaged if that voltage exceeds 5 V. Use a voltage divider to keep the voltage under 5 V. I measured 4.7 V at my rotator. The setup procedure will calibrate your rotator feedback for any voltage below 5 V.

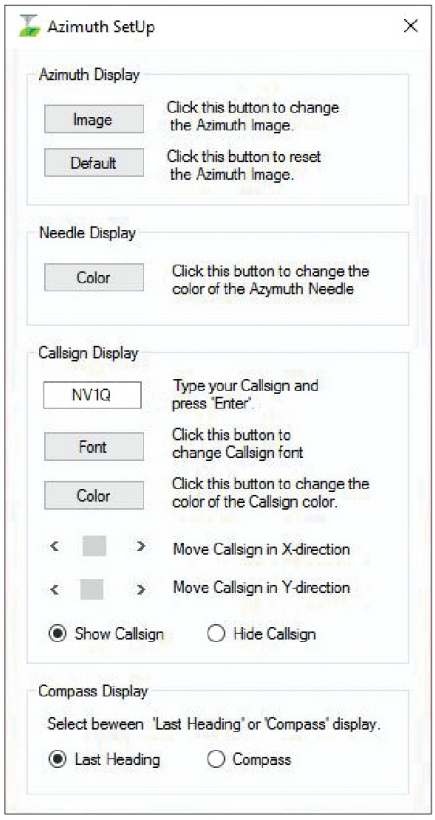

Figure 5 — Insert your call sign, choose the color, and position the call sign on the screen in the azimuth display setup.

A printed circuit board is available from the author. Visit **hcra.org** for updates on this design.

Thanks to John Plaster, K1VOI, for beta-testing this design.

Juergen E. Malner, NV1Q, was first licensed in 1967 as DK1TM, when he was a radio operator in the German Air Force. He earned his Amateur Extra class license in 2013. Juergen received a Diplom-Ingenieur (electrical engineering degree) from the University of Applied Sciences in Bremen, Germany. He retired in 2013, after working 30 years in the paper industry. He enjoys writing software and chasing DX. Juergen lives in Westfield, Massachusetts with his wife, Renate. You can reach Juergen at **nv1q@arrl.net**.

For updates to this article, see the *QST* Feedback page at www.arrl.org/feedback.

New Products

Antenna Insulators from SOTABEAMS

SOTABEAMS offers a line of antenna insulators made from modern lightweight materials. Two types are available: a ribbed center or end insulator, and an open-wire line insulator for high-impedance (600 Ω), low-loss feed lines. The ribbed center or end insulators are 4.7 inches long and weigh about ½ ounce, with a strain rating of 220 pounds. The open-wire line insulators are 4.6 inches long and space the wire at 4.4 inches. Price: ribbed center/end insulator, about $2 each; open-wire line insulator, about $4 for a pack of 10. For more information or to order, visit **www.sotabeams.co.uk**.

ColibriNANO 0.01 – 500 MHz SDR USB Receiver

The ColibriNANO 14-bit direct-sampling, software-defined receiver features specified blocking dynamic range (BDR) of 110 dB, native 0.01 – 55 MHz coverage with up to 500 MHz in oversampling mode, and a switchable 60 MHz low-pass filter (LPF). The radio is compatible with *HDSDR*, legacy *SDR#*, and *ExpertSDR2* software. Supported platforms are *Windows XP* through *Windows 10* and *Linux*, as well as a web client for HTML5 browsers. Price: $300. For more information, visit **www.nsiradio.com**.