Idiom Press Rotor-EZ Rotator Control Protocol

We have received many inquiries regarding the command protocol for our rotator control boards. Both the Rotor-EZ (with serial control option) and RotorCard series control boards use the same protocol, which we call the Rotor-EZ protocol. It has some commands in common with Hy-Gain DCU-1 protocol. If a given software program does not support the Rotor-EZ protocol, usually configuring the program to use the DCU-1 protocol will be sufficient. The Rotor-EZ protocol is required if full control of the hardware configuration via the serial port is desired. Please note that the RotorCard DXA and RotorCard SDX control boards do **not** support the Yaesu protocol.

The Rotor-EZ protocol includes commands that relate to hardware features of the control board. Therefore, not all commands are supported for a given rotator control board. See the table below. Please refer to the rotator control board manual for a complete description of the hardware features.

Important note: all bearing numbers used in Rotor-EZ protocol commands must have three digits. If necessary, the number must be padded with leading zeros. For instance, "`009`" degrees should be used instead of "`9`" degrees.

The RS-232 port talking to the rotator control board should be configured as follows: 4800 baud, no parity, eight data bits, and one stop bit. Some software programs refer to this configuration as "`4800,N,8,1`". Flow control (RTS/CTS) should not be used.

"`AP1xxx<CR>`" sets target bearing to "*xxx*" degrees where *xxx* are three digits between 000 and 360, and executes. For example, a command to rotate to 080 degrees would be "`AP1080<CR>`". Note that the <CR> (carriage return) is often labeled <Enter> on many keyboards.

The complete command set follows. Please note that the commands are case-sensitive, i.e. the "`E`" command is not the same as the "`e`" command. Commands that change a hardware option setting, i.e. `E`, `e`, `O`, `o`, `S`, `s`, `J` and `j`, are effective immediately once the command has been given.

Command	Rotor-EZ	Rotor Card	Description
`AP1xxx<CR>`	☒	☒	sets the target bearing and executes
`AP1xxx;`	☒	☒	sets the target bearing but does not execute
`AM1;`	☒	☒	executes rotation to the bearing set by the "`AP1xxx;`" command
`AI1;` or `AI1<CR>`	☒	☒	inquires current bearing, and responds with "`;xxx`" degrees, where `xxx` is a number between 000 and 360
`;`	☒	☒	terminates rotation
`E`	☒	☐	enable endpoint option
`e`	☒	☐	disable endpoint option
`O`	☒	☐	enable overshoot option
`o`	☒	☐	disable overshoot option
`S`	☒	☐	enable unstick option
`s`	☒	☐	disable unstick option
`V`	☒	☒	reports version number and copyright information
`J`	☒	☐	enable jam protection
`j`	☒	☐	disable jam protection – **not** recommended
`K`	☐	☒	enter RotorCard calibration mode (see manual)
`k`	☐	☒	exit RotorCard calibration mode (see manual)

A Note on the Analog Nature of the Rotators that We Support

We get frequent questions regarding what kind of data to expect when polling the rotator control board for the current bearing. Many of these questions betray a lack of understanding about how the rotator and its control board work. Of course this is often perfectly understandable because programmers write code to drive our rotator control boards without actually having a rotator and control board on hand for testing.

Please keep in mind the basic design of the rotator. The position sensor is a wire-wound potentiometer, which is not very precise. The Rotor-EZ and RotorCard control boards are essentially digital designs controlling a purely analog product. Programming the control board to smooth analog signals and ignore momentary noise and other unavoidable analog inaccuracies was very challenging. Our control boards insulate the programmer somewhat from having to deal with the analog nature of the hardware. Due to practical limits on the price we can charge our customers, we were unable to completely hide analog problems from application developers.

These problems are obvious to anyone who has ever closely watched a Hy-Gain rotator control box in action. The analog position meter frequently jumps around during bearing changes, and often seems to move slightly when the rotator is locked. If you poll a stationary rotator for its bearing via the serial port, the answer will very frequently alternate between two values a few degrees apart. This is a direct result of analog noise.

If this seems frustrating at times, please remember that the rotator itself is the result of design compromises. It was designed to inexpensively point an HF beam antenna with a fairly broad main lobe. Greater accuracy or a "more digital" position sensor would make the rotator more expensive without suiting the majority of the customers.

Frequently Asked Questions

Q. What's the best command to turn the rotator to a new bearing?

A. Either "`AP1xxx<CR>`" or "`AP1xxx;AM1;`" works fine.

Q. What happens if a new bearing command is sent while the rotator is turning?

A. The rotator will stop turning. In the case of the Rotor-EZ, then the rotator will wait five seconds for the brake delay and then drop the brake. The rotator will not turn to the new bearing.

Q. What happens if a new bearing command is sent to a Rotor-EZ board after the rotator has stopped but before the brake is set?

A. Nothing; the command is ignored.

Q. How can I tell what the rotator is doing?

A. The only way to tell what the rotator is doing is to poll the rotator control board with a current bearing inquiry "`AI1;`". If the bearing is changing, the rotator is probably turning. If the bearing is close to the previous target bearing, and has been so for several seconds, then the rotator has probably arrived at the commanded bearing. If the bearing hasn't changed in several seconds but is not close to the target bearing, then the command may have been cancelled by the operator using the front panel controls, or the rotator may have had a hardware failure.

Q. What are some common rotator hardware failure modes to watch for?

A. The rotator might get stuck. The wire-wound potentiometer that serves as a position sensor might be so dirty that the rotator control board can't smooth the signal enough to determine the current bearing; this problem is often indicated by a "jumping needle" on the analog bearing meter.

Q. What's up with the brake and the brake delay on Rotor-EZ-equipped Hy-Gain rotators?

A. The Rotor-EZ board automatically waits five seconds after turning, and then applies the brake. During the time that the brake is off, the antenna system may be reverberating or coasting. For the health of the antenna system and the rotator it is recommended to wait several seconds after the rotator has apparently arrived at the target bearing before turning to a new bearing.

Q. Do I have to worry about the brake and the brake delay on RotorCard-equipped Yaesu rotators?

A. No. The Yaesu rotators do not have brakes like the Hy-Gain rotators, and there is no brake delay.

Q. I need to know more about the Rotor-EZ endpoint option / overshoot option / unstick option / jam protection option.

A. That information is in the Rotor-EZ manual, which is available upon request. One thing to keep in mind: if you override a jumper setting with a serial port command, the changed setting will only be in effect until the unit is switched off. The next time the unit is powered on, it will take its settings from the hardware jumpers again.

Q. Does the programmer have control over the fast speed / slow speed mode of the Yaesu rotators?

A. No. However the RotorCard control board automatically starts the rotator turning in the slow speed, then switches to high speed for most of the rotation, and finally switches to low speed just before the rotator arrives at the new bearing.

Q. Has anyone shared any code to control rotators equipped with Rotor-EZ or RotorCard control boards?

A. Not yet. If you would consider sharing your code, please contact us.

Q. I'm having trouble getting the rotator to stop within 1° of the bearing I ordered via a serial port command.

A. Please keep in mind the basic design of the rotator. Exact to-the-degree control of the rotator is not possible given the rotator's design compromises. Remember, the rotator was designed to inexpensively point an HF beam antenna with a fairly broad main lobe. Greater accuracy would make the rotator more expensive without suiting the majority of the customers.

Q. Could you give me some suggestions about how to handle the pointing inaccuracy?

A. We suggest that you consider the rotator to have arrived at the target bearing when it is within "*z*" degrees of the target. You must experiment to determine what value of "*z*" works for your application. There may be variation from one rotator to the next.

Q. How do I parse the data stream coming back from the rotator control board?

A. The data stream is pretty simple to parse. There are only two commands that return anything via the serial port, "**V**" and "**AI1;**" (or "**AI1<CR>**"). "**V**" as mentioned earlier returns "**;xxx**" degrees, where **xxx** is a number between 000 and 360. "**AI1;**" (or "**AI1<CR>**") returns a version information string. Responses are not delimited on either end. Other than answering those two commands, the rotator control board sends nothing out the serial port.