K3NG Arduino Amateur Radio Rotator Controller http://blog.radioartisan.com/yaesu-ro...

arduino amateur-radio rotator-controller antenna-tracking

README.md

K3NG Rotator Controller

Introduction

This is an Arduino-based rotator interface that interfaces a computer to a rotator or rotator controller, emulating the Yaesu GS-232A/B and Easycom protocols which are supported by a myriad of logging, contest, and control programs. It can be easily interfaced with commercial rotator control units. With the addition of a proper capacity power supply and several interface components such as relays, this unit could also serve as a total replacement for a rotator control unit or serve as the basis for a 100% homebrew rotation system. Several azimuth and elevation position sensors including potentiometers, rotary encoders, and I2C devices are supported. The code is very flexible, modular, and easy to read allowing intermediate and advanced experimenters and builders to customize it.

Documentation

Full documentation is located here. Please read it! Volunteers for maintaining documentation are needed.

Features

- Azimuth only and azimuth / elevation rotator support
- Serial interface using the standard Arduino USB port
- Control Port Protocol Support:
- Yaesu GS-232A & GS-232B
- Easycom
- Support for position sensors:
- Potentiometers / Analog Voltage
- Rotary Encoders
- Incremental Encoders
- Pulse Output
- HMC5883L digital compass
- ADXL345 accelerometer
- LSM303 digital compass and accelerometer
- HH-12 / AS5045
- A2 Absolute Encoder (under development)
- LCD display (2 or 4 rows, at least 16 columns)
- Can be interfaced with non-Yaesu rotators, including homebrew systems
- Directional indication on LCD display (North, South, North Northwest, etc.) along with degrees
- Intelligent automatic rotation (utilizes overlap on 450 degree rotators)
- Support for both 360 degree and 450 degree azimuth rotators or any rotation capability up to 719 degrees
- North Center and South Center support
- Support for any starting point (fully clockwise)
- Optional automatic azimuthal rotation slowdown feature when reaching target azimuth
- Optional rotation smooth ramp up
- Optional brake engage/disengage lines for azimuth and elevation
- Buttons for manual rotation
- Command timeout
- Timed interval rotation
- Overlap LED Indicator
- Help screen
- Speed Control, both single PWM output (compatible with Yaesu controllers) and dual PWM rotate CW and rotate CCW outputs and dual elevate up and elevate down outputs
- Variable frequency outputs
- Preset Control using either potentiometers or rotary encoders with optional preset start button
- Speed Potentiometer
- Manual Rotation Limits
- Classic 4 bit, Adafruit I2C LCD, and Yourduino.com Display Support
- Optional tenth of a degree support with Easycom protocol (i.e. 123.4 degrees)
- Park button
- Azimuth and elevation calibration tables
- Host unit and Remote unit operation for remotely located sensors using two Arduinos or ATMega chips
- Works with hamlib rotctl/rotcltd, HRD, N1MM, PST Rotator, and many more programs
- Moon and Sun Tracking
- GPS Interfacing
- Realtime Clock Interfacing

Acknowledgements

John, W3SA, has tested on a Yaesu Az/El unit, contributed several updates to the elevation code, and tweaked the code for a 16 column LCD display.

Anthony, M0UPU, wrote about his rotator controller construction and is offering PC boards.

Bent, OZ1CT, has contributed several ideas and feature requests, and performed testing.

G4HSK has a nice page documenting his project using this code, the PstRotator control software, and a Yaesu G-5500 rotator.

DXpeditions

I will donate parts, units, or specially customized software for DXpeditions. Email me at anthony dot good at gmail dot com. DX IS!

Support and Feature Requests

Please consult this page for support information. Feature requests and bugs are documented and tracked on GitHub.

Please note that I do this work in my spare time as I can and I am not a professional developer, however I play one on TV. I do my best to answer support requests, however I don't like having to answer questions for items that are explained in the documentation. I do maintain a list of feature requests. Development items are prioritized by me based on the level of difficulty and what I'm interested in. I welcome code contributions, code testing, bug reports, and any help you can provide. This can even be helping with documentation or providing support to others on the Radio Artisan discussion group.