Satellite Tracker Interface

By Mark Spencer, WA8SME
Former ARRL Education & Technology Program Coordinator
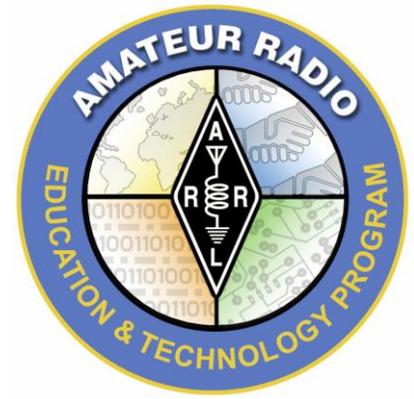
Bringing space into the classroom is an incredibly powerful learning experience for students. The marriage between wireless technology literacy and space literacy is a strong one that produces a portfolio of activities that can engage virtually all students at some ability and interest level.
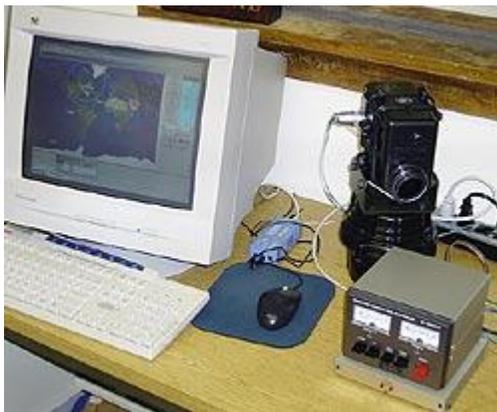
This portfolio of activities can range from simply having a satellite predication software package running in the back of the classroom on an old, salvaged computer to a full-up ground station to communicate with the astronauts in the International Space Station through the ARISS program. Planning and preparing to bring space into the classroom is hard work, and it can be expensive depending on the level of material and experiences that the teacher wants their students to share.

One of the station requirements to be scheduled for an ARISS contact is to have and antenna system consisting of an OSCAR style circular polarized beam antenna and an azimuth/elevation rotor control. Because of the orbital characteristics of the ISS, manually controlling the azimuth/elevation rotor system would be difficult at best and could jeopardize the success of the contact. Consequently, a computer controlled rotor system would be preferable (not only for an ISS contact but for all low orbiting satellite work). The only azimuth/elevation rotor system for the ham radio market is the Yaesu G-5500. With a price tag of about $600, this is an expensive package. The rotor to computer interface accessory is an additional $500. In this article I describe a simple rotor to computer interface that can be built for a fraction of the cost of the commercial accessory. Not only will the builder save the cost difference that could be applied to other station enhancements, but when you build your own, you will learn volumes about orbital mechanics, antenna orientation strategies, and you will be familiar enough with the inherent software of the interface to make adjustments as needed to keep the antenna pointing system in peak condition.

This particular project is designed to interface the Yaesu G-5500 rotor system to an IBM computer running the NOVA satellite tracking software. This project is by no means unique, numerous interface designs can be found on the web (This circuit design and software is an adaptation of a similar system developed by Gene Brigman, KC4SA, and Mark Hammond N8MH. Specifics of their system can be found at:
www.webpages.charter.net/mlhammond/ham/SAEBRTrack/). This project however is a no frills design that does not duplicate the antenna pointing indications that are available from the G-5500 console (therefore reducing costs). Also the programming is straightforward and "airy" allowing for lots of modifications to suit individual applications. The NOVA satellite tracking software package was chosen because it contains many powerful features for not only pointing antennas but also display options that are invaluable to the teacher instructing orbital fundamentals (I particularly like the map graphics and the 3D global view options). Specifics of the NOVA software package can be found at: www.nlsa.com/
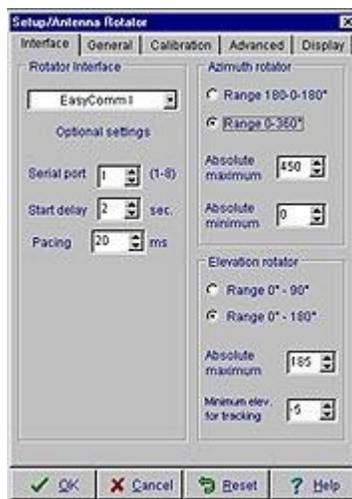
NOVA data output architecture. Along with displaying satellite position and antenna pointing information on the computer screen, NOVA also sends the antenna pointing information to the serial port using the EASYCOM1 protocol (more information at:
www.mustbeart.com/software/easycomm.txt). Rotor parameters and limitations are set into drop-down user interface screens so that the rotor positioning is adjusted to the particular rotor capabilities. The antenna rotor setup for this interface should be as illustrated in the accompanying illustration (set the appropriate com port).

G-5500 architecture. Four momentary switches control the G-5500 azimuth/elevation rotor system: up, down, clock-wise, and counter-clock-wise. The G-5500 rotor system has variable resistors connected to the azimuth and elevation rotors that are turned along with the antenna. These resistors produce a variable voltage within the rotor control box from 0 volts at 0° to a user set maximum, in this case 4.5 volts at either 450° for azimuth or 180° for elevation. The control switches and antenna position voltages are available through an 8-pin external control jack on the G-5500 console. Set up your rotor as detailed in the instruction manual so that at maximum azimuth and elevation the voltages are 4.5 volts.

Interface functions. As detailed in the circuit diagram, there are two primary components in the interface; the BASIC Stamp micro controller and a 12-bit analog to digital converter (ADC). The satellite interface uses a 12-bit analog to digital converter (ADC) to read the position voltages from the G-5500 and converts them to a digital word value between 0 and 4096. Satellite position information is in degrees, antenna position information is in a digital word, one needs to be converted so that the BASIC Stamp micro controller can compare the two values and send appropriate commands to move the rotors. The BASIC Stamp uses the ADC to read the antenna position from the G-5500, and used the antenna position information sent from NOVA, converts the number of degrees into a scaled digital value, and then compares the digital values (where it is supposed to be compared to where the antenna is). Finally, the BASIC Stamp determines which direction to turn the antenna; up/down, left/right, and closes appropriate relays that are connected in parallel with the direction control switches in the G-5500 console to turn the rotors. The relays remain closed until the antenna is detected to be in the correct position (through reading the ADC). NOVA generated satellite position data is then monitored for the next antenna movement command.

When am I ever going to use this stuff? I don't know how many times I have heard this lament during my teaching career. Well, to understand the conversion of degree data to scaled digital data, you have to use basic algebra skills. Here is a use for this stuff!

One complicating factor in making this conversion is that the BASIC Stamp is limited to integer mathematics. This means that even for simple mathematical operations, decimal parts of the computations are truncated and the resulting answers are rounded down to the nearest integer. This can inject unacceptable errors into computations. I will deal with the integer math limitations later.

To make the conversion from degrees to digital, you will have to dust off some basic algebra skills for linear equations. The output of the ADC is linear and follows the algorithm:

$$Y = mX + b$$

Where m is the slope of the line ($\frac{Y_2 - Y_1}{X_2 - X_1}$) and b is the Y-axis intercept. In terms of satellite position in degrees and digital word, Y is the converted value in digital word and X is the position in degrees. By using the little program called ADC_Checkout.bs2 (available from the author), you can determine the digital word value for the end points (0° and 450° or 180° as appropriate) of the rotor. Once armed with these two data points, you can determine the slope and Y intercept values for the two equations needed to convert the azimuth and elevation degrees into digital words.

The following computation example uses the data gathered with the prototype satellite tracker interface. You should not need to go through these steps to calibrate your interface, this is provided only for those who have the desire to know.

Satellite Tracker Interface ADC Data			
AZ Degrees		AZ Digital Word	
X1	0	Y1	30
X2	450	Y2	3710
EL Degrees		EL Digital Word	
X1	0	Y1	12
X2	180	Y2	3712

Azimuth: slope $m = \dfrac{3710 - 30}{450} = 8.18$ Y- intercept b = 30

Elevation: slope $m = \dfrac{3712 - 12}{180} = 20.56$ Y-intercept b=12

Because the BASIC Stamp uses integer math, if we were to multiply the number of degrees by these slope values, 8.18 or 20.56, the fractional part would be truncated and result in error. The BASIC Stamp has a math operator that takes care of this situation, the Multiply Middle operator (*/). The Multiply Middle operator multiplies variables and/or constants, returning the middle 16 bits of the 32-bit result. This has the effect of multiplying a value by a whole number and a fraction. The whole number is the upper byte of the multiplier (0 to 255 whole units) and the fraction is the lower byte of the multiplier (0 to 255 units of 1/256 each). The */ (starslash) instruction gives you an excellent workaround for the BASIC Stamp's integer-only math. It is helpful to use hexadecimal notation for the multiplier so that you can keep track of the upper (whole number) and lower (fractional part) byte. You can use the WINDOWS calculator in the scientific mode as a convenient way to convert decimal numbers into the hex equivalent.

For example, to convert the azimuth value of degrees into the ADC digital word equivalent we need to multiply the number of degrees by the slope of the line and add the y intercept.

$$Y = 8.18X + 30$$

$8 = 08$ in hex. $.18 * 256 = 46.08$ rounded to $46 = 2e$ in hex. Multiplier in hex is $082e therefore: $Y = \$082e * X + 30$. The line of code to reflect this algorithm is: **new_az=new_az*/$082e+30**. In a like manner, the multiplier for the elevation conversion is $148f (20 decimal in hex is 14). The line of code to reflect this conversion is: **new_el=new_el*/$148f+12**.
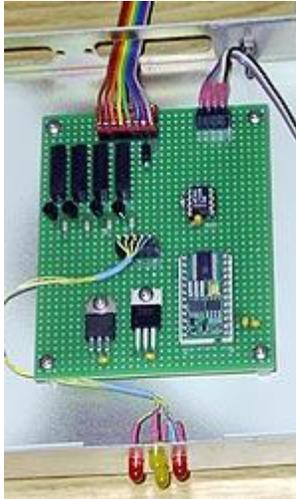
Construction. The layout and construction of the satellite interface is not critical. As shown in the photo, the interface is constructed on a standard prototyping board using point to point wiring. Wiring the serial port as indicated in the circuit diagram will allow the same computer to be used to program the BASIC Stamp and to access the serial data from NOVA. I arranged the LED display in a pattern to give a visual indication of rotor direction. The power for the interface comes from the G-5500. This voltage source is around 14 volts with no load, but when the rotors are engaged, there is a significant voltage drop. To provide some level of stability, the voltage from the G-5500 is run through the 12-volt regulator (the power source for the relays) before the voltage is further reduced and regulated by the 5-volt regulator to power the BASIC Stamp and ADC.

I suggest the interface be constructed and tested in stages. Begin first by wiring and testing the voltage regulators. Next add the BASIC Stamp and serial port. The BASIC Stamp is programmed using the BASIC Stamp editor program available from Parallax at: www.parallax.com/html_pages/downloads/software/software_basic_stamp.asp.

You can test this portion of the circuit by using the features of the editor.

Next wire-up the ADC and the position voltage connections to the G-5500. Before you test this portion of the circuit, ensure that you set the full-scale voltage for the 180° and 450° rotor positions to 4.5 volts as outlined in the G-5500 manual. Run the program ADC_Checkout.bs2 to test the ADC portion of the circuit. Note the ADC values as you rotate the rotors through their full travel, the values for 0° and 180° and 450° should be close to the values listed in the table above. If they are, you should be able to use the main software without adjustment, if not, and the wiring is correct, you will have to do a little math and program modifications as detailed above.

Finally wire up the relays and do the final smoke check with the main program (Sat_Tracker_Final.bs2) up-loaded to the BASIC Stamp.

The software. The real advantage to making your own interface, aside from the obvious fact of reduces costs, is that you have control of the interface through the software. The software is well

documented and should be easy to follow. If the ADC values for full scale are significantly different from the values listed in the table above, you will probably want to make adjustments to the constants in the conversion algorithms. Also, if the rotor system chatters in operation due to antenna torque movement, adjust the tolerance value to alleviate the problem.

Operation. Connect the interface to the G-5500 and to the computer serial port. Run the NOVA software; select the satellite to be tracked, select auto-tracking and the rotors will be commanded to the position for the start of the next pass. When the satellite is within range, the tracker will respond by updating the rotor position until the pass is completed. Aside from manipulating the NOVA software, the satellite tracker interface should take care of the rest.

Final thoughts. For the system to perform successfully, you will have to become familiar with the G-5500 rotor system, the operation of the NOVA satellite tracking software, and the BASIC Stamp editor for programming the Stamp. They all come together in the interface. So this project goes beyond simply wiring a duplicate circuit, that's the easy part. The fun part is learning how to make all the components of a satellite antenna tracking system work together.

Though this project specifically focused on the G-5500 and NOVA components, the interface could easily be adapted to other rotor configurations. It would just take a little homework to adjust the interface software.