Simple Satellite Tracker Interface for Yaesu G5500 or G5400 Rotors

by Mark Spencer, WA8SME, mspencer@arrl.org

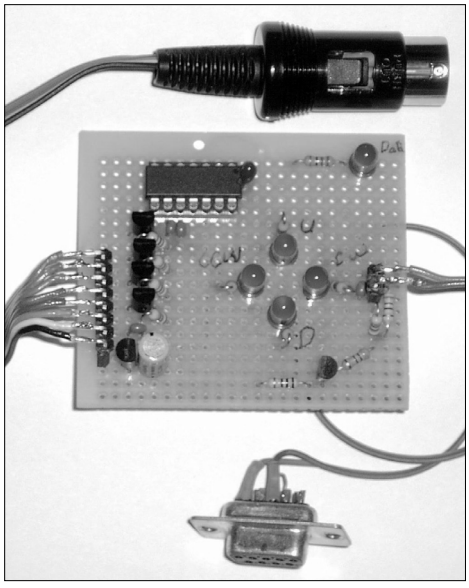
In the September 2005 QST article "A Satellite Tracker Interface", I described a simple interface that connects the Yaesu G5500 rotor system to a computer running either the NOVA[1] or SatPC32[2] satellite tracking software to automatically point satellite antennas. My motivation for that project was to illustrate a relatively inexpensive interface system that schools could duplicate to satisfy one of the station requirements to qualify for an ARISS contact. I also wanted to illustrate a project that hams and students alike could use to learn about microcontroller programming. Readers of that article suggested two things:

1. The interface should also work with the Yaesu G5400 rotor system.
2. Make the interface even less expensive.

In this article I describe a second version of the interface to address those two suggestions. I also had the self-serving goal of meeting a new programming challenge and learning experience. The result described here is an interface that costs approximately $20.00 in parts (Table 1) and will work with both the G5500 and G5400 rotors. The hand wired circuit is shown in Picture 1.

Hardware

The circuit depicted in Figure 1 is based on the 16F688 PIC (programmable interface controller). The 16F688 features a universal asynchronous receiver transmitter (USART) for serial communications, 10-bit analog to digital converters (ADC) for reading rotor position voltages and 12 input/output (IO) ports for control signals. The voltage source for the interface comes from the rotor controller and is regulated down to 5 volts by the 78L05 regulator. Transistor Q1 converts the RS232 voltages from the computer serial port to TTL levels needed to communicate with the PIC. The transistors Q2 through Q5 do the switching of the rotor direction control relays in the rotor controller. The LED's in the circuit are not required for proper circuit operation but do provide helpful information when the interface is operating. The capacitors C4 through C6 provide some filtering of noise on the RS232 and the ADC lines. Of course, it all is controlled by the software inside the PIC.



Picture 1: Tracker hardware constructed in breadboard format.

Software

Programming the PIC in assembly language certainly met my goal of a programming challenge, particularly doing the required binary math to convert degrees into equivalent ADC readings. The software is available to anyone for the price of an e-mail request to mspencer@arrl.org. I have attempted to liberally comment the software to allow the reader to follow the logic. The following are the basic steps accomplished to turn a command from the tracking software to point the antenna into the actions required to move the antenna:

1. Read the antenna position commanded by the NOVA or SatPC32 software. These software packages use the EASYCOM1 [3] protocol. Table 2 is an example of the ASCII command strings sent by the software packages.
2. Convert the commanded antenna position in degrees to equivalent position voltage in digital form.
3. Read the current antenna position with the ADC.
4. Compare the desired position with the current position.
5. Determine the rotor movement direction to correct for position error (if any).
6. Move the rotors.
7. Keep reading the antenna position and continue to move the rotors until the desired antenna position is achieved.
8. Start the process over again by reading the new position command from the controlling program.

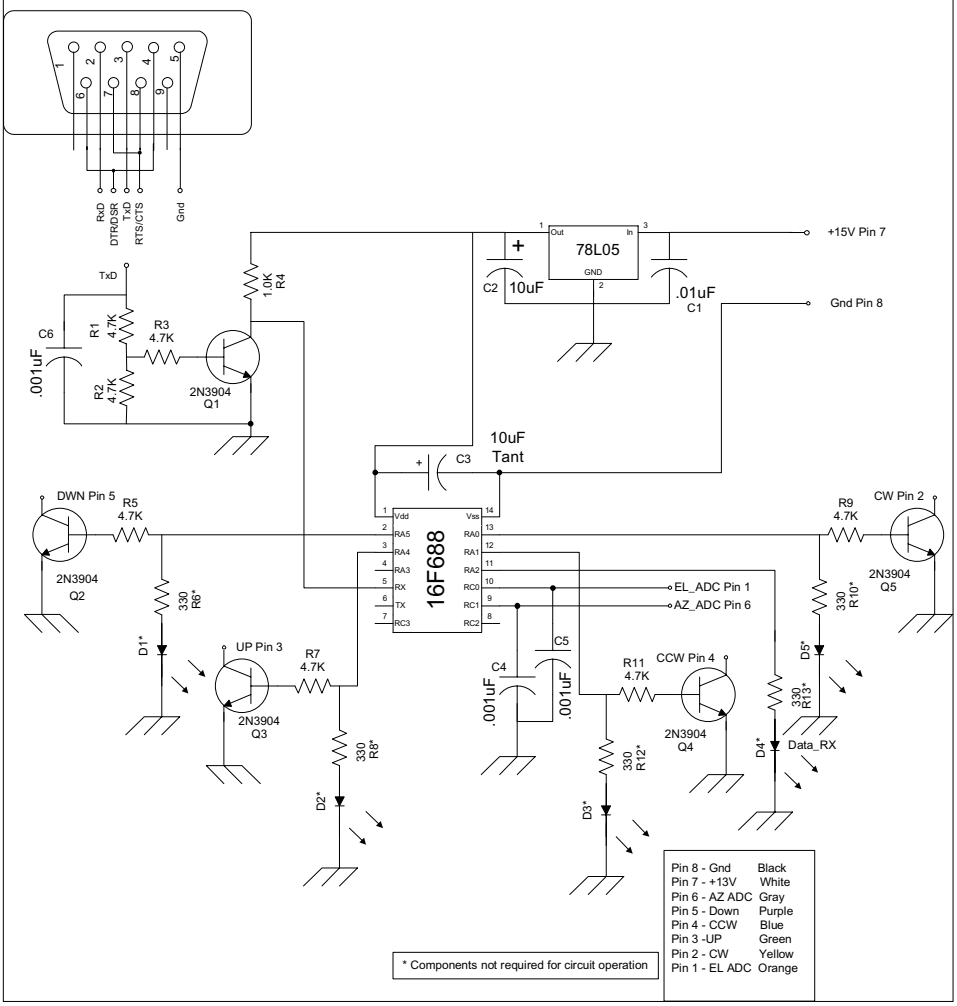
Operation

The interface set-up and operation is pretty transparent. The PIC is programmed to operate at the standard of 9600 Baud, so the NOVA and SatPC32 should be set up for the appropriate COM port and baud rate. Set NOVA to use the EasyComm 1 rotor interface. Set SatPC32 to use the SAEBRTrackBox rotor interface. Set the rotor azimuth stop position (north or south)

Table 1: SatTracker 688 Parts List

Number Required	Part Designation	Part Description	DigiKey
1	U2	PIC	PIC16F688-I/P-ND
1	U1	78L05 Regulator	296-1365-1-ND
5	D1, D2, D3, D4, D5	Red LED	67-1105-nd
5	Q1, Q2, Q3, Q4, Q5	2N3904	2N3904FS-ND
5	R6, R8, R10, R12, R13	330 Ohm	P330BACT-ND
1	R4	1K Ohm	P1.0KBACT-ND
7	R1, R2, R3, R5, R7, R9, R11	4.7K Ohm	P4.7KBACT-ND
1	C1	.01uF	P4582-ND
1	C2	10uF Electrolytic	P1176-ND
1	C3	10uF Tantalum	478-1840-ND
3	C4,C5, C6	.001Uf	P4551-ND
1	J3	9-Pin Serial	A2100-ND
1	J1	6-Pin Female	WM2004-ND
1	J2	8-Pin Female	WM2006-ND
1	J1A	6-Pin Male	WM4304-ND
1	J2A	8-Pin Male	WM4306-ND
14		Pins	WM1114-ND
1	Socket	14-Pin DIP	ED3114-ND

Satellite Tracker Interface Using PIC16F688
for
Yaesu G-5500 or G5400 and NOVA or SatPC32
Figure 1



as appropriate for your rotor system. Set the elevation stop position to 0° and 180°. Adjust the software pacing to meet your particular operating conditions. I can provide screen capture documentation to help you set up your NOVA or SatPC32 software upon request.

When you select a satellite for tracking and turn on your rotor controller, you will see the data LED light up when data is being received from the controlling program. When the data light goes out, the direction LED's will light in-turn to indicate that the signals to close the appropriate rotor control relay are being sent to the rotor controller (you will hear the relays close with a audible click). When the antenna has reached the commanded position, the LED's will go out and the relays opened. This process will continue for the duration of the satellite pass.

So if you are looking for a simple project to build to enhance your satellite operation, if you are looking for a programming challenge or if you just wanted to beat the high price of a commercial controller interface, this project might be for you. I encourage you to contact me if you have questions or comments. E-mail me at mspencer@arrl.org or phone 530-495-9150 (PST), or mail at 774 Eastside Rd., Coleville, CA 96107.

References

[1] www.nlsa.com/

[2] www.dk1tb.de/

[3] www.mustbeart.com/software/easycomm.txt

Table 2: Program Command Outputs
NOVA:
AZ317.4 EL180.0 UP3.56630275 XXX DN000000000 XXX
SatPC32:
AZ360.0 EL000.0