Satellite CAT Interface for Working the Analog Birds

Expand your satellite operations from FM to SSB and add a new dimension to your skill set.

Mark Spencer, WA8SME

FM amateur satellites, basically repeaters in the sky, have proven to be very popular. They are easy to access with standard dual band FM radios, a simple directional antenna, a lot of listening and a lot of practice. There are times, however, when the FM birds have become the victims of their own success. The declining number of these high altitude, single channel repeaters has resulted in overcrowding and fierce competition for contacts, marginalizing many who have enjoyed operating the FM satellites, particularly low power and portable stations.

Expand Your Satellite Operations

An alternative to the FM repeating satellites are the wideband analog birds that retransmit a whole sub band of SSB and CW signals. These birds are equally capable and instead of a single channel they use transponder technology that transmits a wider band of frequencies in which multiple contacts can be retransmitted simultaneously (each signal sharing the available bandwidth and RF power of the satellite). The use of transponders and SSB/CW, however, requires more precise frequency control to find the other station, and then to accurately track the other station with its Doppler shift — not a major concern with the FM birds.

Software Brings it All Together

Current satellite tracking software not only tracks the location of the satellite, but also can tune the radio to provide Doppler shift frequency correction (computer aided tuning or CAT). While this helps, it is not essential for operating through the analog birds. It can be argued, however, that being able to hear the downlink while transmitting (full duplex) is critical to successful operations on the analog birds. While the wideband analog satellite alternatives to the FM birds have a lot of capabilities, they are underutilized. There have been many passes on the analog birds when I have been talking to myself while I have been unable to get a word in edgewise on the next FM bird pass. What gives?

I think the answer includes:

- The lack of readily available (both functional and affordable) all-mode, dual band VHF/UHF, full duplex radios.
- It takes more operator skill and practice to operate the analog birds once the investment is made in the equipment.

Shown above is Teachers Institute Instructor Matt Severin, N8MS, communicating through a wideband satellite.

The alternative to an expensive full duplex radio is obtaining two all band transceivers (or a transverter for the downlink in combination with an HF rig for the IF), one for the uplink, the other for the downlink thereby achieving full duplex. Admittedly this is an expensive alternative.

An Interface to Bring It All Together

After this lengthy introduction, I come to the point of this article. What follows is a description of an interface project that will allow you to operate the analog birds with a single non full duplex, all-mode VHF/UHF transceiver. While the circuit and the software are designed for a specific radio (the popular, compact, multimode, Yaesu FT-817 HF/VHF/UHF portable transceiver) and satellite tracking software package (*SatPC32*), it serves as an example that can be adapted to other radios and other tracking software — left only to the imagination of the builder. This project is not, however, the ultimate alternative to a full duplex all mode transceiver setup, but it will facilitate your exploration of the analog birds more affordably.

First Pick a Radio

I evaluated the suitability of a number of radios for use by teachers to bring space technology into their classrooms while developing the *ARRL Education and Technology Program Teachers Institute-2 Space* (TI-2) course. The rig that floated to the top was the FT-817. This is a relatively affordable low power, all band, all mode transceiver that can be battery operated or operated from a fixed power supply. What

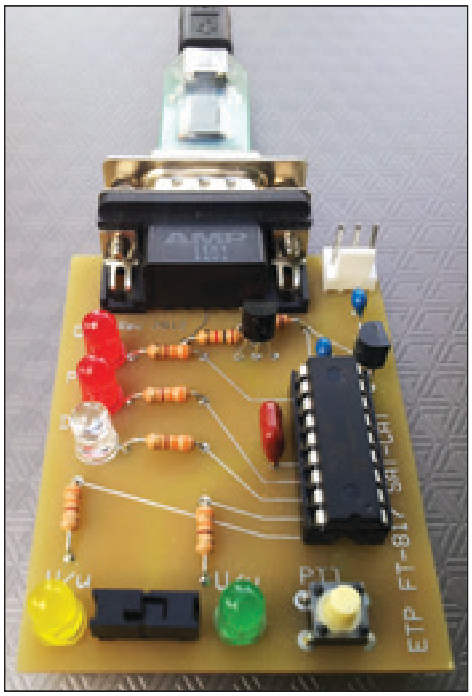

Figure 1 — The Sat CAT Control Interface prototype for use with a Yaesu FT-817.

attracted me most to this transceiver is that it can operate in all modes, has split frequency capabilities (dual VFOs), can be CAT controlled and can operate crossband in the VHF and UHF bands. It is not, however, a full duplex radio.

I apparently am not alone in my favorable opinion of this radio as there are a number of satellite operators who are using this rig (some use two FT-817s to allow full duplex operation). During the TI-2, the teachers use the FT-817 to copy the CW telemetry transmitted by many of the CubeSat satellites, to monitor the analog birds and to make contacts through the FM birds.

Then Pick the Software

The satellite tracking software package that I adopted for the TI-2, as well as for my personal use, is *SatPC32* (there are many other equally capable software packages). *SatPC32* tracks the location of the selected satellites the operator wants to monitor, sends out antenna positioning commands that can be used with a rotator control interface to keep directional antennas pointed at the birds (numerous rotator control interface circuits have been published), sends out radio specific CAT commands to tune the connected radio transceiver to the operating frequencies and keeps those frequencies on track for Doppler correction.

It sounds like *SatPC32* should be able to make a non-full-duplex transceiver work for the analog birds. I have found that the combination of single FT-817 and *SatPC32* is lacking in some respects. The problem is that when updating the receive and transmit frequencies, *SatPC32* toggles between the VFO A and VFO B functions of the radio. This action, which chops up the reception, makes the radio unusable. Because the frequency updates happen regardless of the status of the PTT of the radio, there is a possibility that the operator will key up to transmit while the transceiver is on the VFO assigned for receive. This could blow out a receive preamplifier if one is used. *SatPC32* can control two radios at one time so it works perfectly with two FT-817s, one for uplink, the other for downlink. But with one FT-817 it is lacking.

Enter the Sat CAT Control Interface

The FT-817 Sat CAT Control Interface (prototype shown in Figure 1) described here is designed to intercept the radio tuning commands being sent by *SatPC32* and allow only the receive frequency updates to be passed to the transceiver while in the receive mode and to pass only the updated transmit frequency while the rig is transmitting. This prevents the constant toggling between VFO A and VFO B while receiving, making reception normal and prevents the inadvertent transmission on the receive frequency during

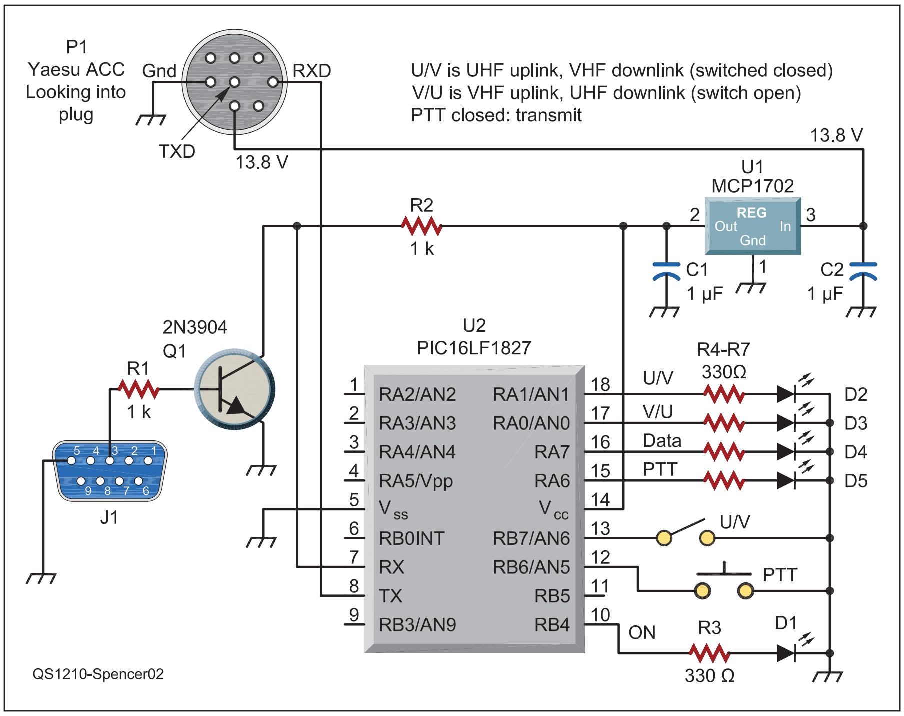

Figure 2 — Schematic diagram and parts list for the interface hardware, based on the PIC16LF1827 microcontroller. Digi-Key parts are available from **www.digikey.com**.

C1, C2 — 1 µF, 50 V electrolytic capacitor (Digi-Key 445-2865-ND).
D1-D5 — LED [Digi-Key 67-1105-ND (Red), 67-1112-ND (Green)].
J1 — 9-pin D-sub jack (Digi-Key A35107-ND).
P1 — Plug to match radio ACC jack (Digi-Key CP-2080-ND).
Q1 — 2N3904 or equivalent transistor (Digi-Key 2N3904FS-ND).
R1, R2 — 1 kΩ, ¼ W resistor.
U1 — MCP1702 3.3 V regulator IC (Digi-Key PIC16LF1827-I/P-ND).
U2 — PIC16LF1827 microcontroller (Digi-Key MCP1702-3302E/TO-ND).

a contact. The interface consists of two parts, the hardware and the software.

Control Interface Hardware

The interface hardware (Figure 2) is based on the PIC16LF1827 microcontroller. This device has USART functionality that makes it easy to connect the microcontroller to other devices serially. The output of the computer running *SatPC32* is at serial RS232 levels (±12 V) while the input to the PIC16LF1827 is at transistor-transistor logic (TTL) voltage levels (3.3 V), so some voltage level conversion is required. This is accomplished by the 2N3904 NPN transistor and current limiting resistors. The CAT input to the FT-817 is at TTL levels, so no voltage conversion between the PIC and the radio ACC connection is needed. There are a number of indicator LEDs mounted on the board as labeled in the circuit diagram for operator convenience and for operation and mode verification. The PTT switch of the circuit is used instead of the microphone PTT switch on the FT-817 microphone.

Control Interface Software

The real meat of the project is in the microcontroller software. The microcontroller software for this project is written in the C programming language and is available for download on the QST-in-Depth website as well as from the author upon e-mail request.[1] Before the logic of the microcontroller software can be discussed, a review of the output CAT command format from *SatPC32* is in order. This would be a good time to dig out the *FT-817 Operator's Manual* and review pages 70-73. (Those who are adapting this project to a different transceiver will have to study their specific radio CAT command format and author software to match.)

To control the FT-817 via a computer, specific commands in a specific format must be transmitted by the computer to the radio ACC jack through a CAT interface. The Yaesu interface is the CT-62 option — basically a voltage level converter between RS232 and TTL levels. These commands are contained in a 5 byte structure with the individual bytes sent LSB first, MSB last at user specified baud rates (this interface is based on 9600 baud). The first 4 bytes of the command structure are parameters (data such as frequency to be set) or dummy place-fillers and the last byte is the actual command *Opcode* as shown in Table 1.

The decimal values of the parameters are

[1]**www.arrl.org/qst-in-depth**

Table 1
Data Command Structure

Parameter				Opcode
P1	P2	P3	P4	CMD

constructed in binary coded decimal (BCD) format with the first nibble (4 bits) of the byte containing the first decimal digit (0-9), and the second nibble of the byte containing the second decimal digit. The combination of the two nibbles, a byte, in turn is converted to a single decimal value for transmission. While this can be intimidating for the computer novice, my point is to illustrate that it will take some thought and study to understand the formats used by both the controlling software and the radio being used.

Here is an example. Let's assume the controlling software is sending a frequency of 435.345 MHz to the radio. The first two decimal digits of the frequency (43) are converted into BCD. The binary code for 4 is 0100. The binary code for 3 is 0011. Putting the first decimal digit into the high nibble and the second decimal digit into the low nibble creates the byte 0100 0011 or 0x43 hexadecimal or 67 decimal (0x## indicates hexadecimal). You can use the scientific calculator that is an accessory to *Windows* to convert from one numbering system to another. I use it all the time.

A sample of the relevant FT-817 commands for this project is contained in Table 2. The command format to set the frequency to the radio is P1:P2:P3:P4:01 where the P#'s make up the frequency in BCD, the 0x01 is the hexadecimal Opcode to set the frequency.

Table 2
Data Transfer Example

Set Freq Hex				Opcode
0x43	0x53	0x45	0x00	01
Set Freq Decimal				**Opcode**
67	83	69	00	01
Upper Nibble		***Lower Nibble***		
Decimal	1	4		
Hex	0x1	0x4		
Binary	0001	0100		
Byte	20			
Parameter				**Opcode**
P1	P2	P3	P4	CMD
Set Freq				**Opcode**
0x14	0x59	0x17	0x00	0x01
Toggle VFO A/B				0x81
PTT ON				0x08
PTT OFF				0x88

Table 3
***SatPC32* CAT Command Set with FT-817 Radio for VO52 Satellites**

Command	Function
0000000002	WAKE UP
0000000081	TOGGLE VFO
8A0000000A	CTCSS OFF
0000000081	TOGGLE VFO
00000000F7	READ RX STATUS (NO EQUIPMENT ATTACHED)
1459170001	RX FREQUENCY
0000000081	TOGGLE VFO
4352449201	TX FREQUENCY
0000000081	TOGGLE VFO
0152449207	OPERATING MODE
0000000081	TOGGLE VFO
0152449207	OPERATING MODE
1459170001	RX FREQUENCY
0000000081	TOGGLE VFO
4352449201	TX FREQUENCY
**********	SEQUENCE CONTINUES
0000000081	TOGGLE VFO
8080808080	*SatPC32* TRACKING OFF, CMD LOCK OFF

Translating this into hexadecimal the five byte command would be: 0x43 0x53 0x45 0x00 0x01. The decimal values of the five byte sequence would be: 67 83 69 00 01.

The other relevant commands needed to control the FT-817 for this project are TOGGLE VFO, PTT ON and PTT OFF.

Let's now take a look at the command set that is conveyed to the FT-817 from *SatPC32*. Table 3 is a sample.

Notice that the program commands the transceiver to toggle the VFO and then sends the frequency. There are other commands that set the operating mode. This is why *SatPC32* doesn't perform well with a single FT-817. All of these commands also complicate the software for the interface. The command set for an older Yaesu radio, such as an FT-736, is similar to the command set for the FT-817, but it is also simpler. Table 4 is a sample of the command set for the FT-736.

In this case, the *SatPC32* software only sends the alternating transmit and receive frequencies and does not send commands to toggle the VFO (this is because the older FT-736, an early satellite capable transceiver, was full duplex capable with two independent VFOs and therefore did not need to be toggled). You can sample the commands coming from *SatPC32* with a terminal program such as *HyperTerminal* or *Putty*, but you need to do some conversion between numbering systems. The command set transmitted by *SatPC32* is in decimal format. The number 1 is the actual decimal value for number 1, the number 127 is the actual decimal value for 127.

Terminal programs use the ASCII format to represent the character being transmitted. So to display the number 1 in the HyperTerminal screen, the transmitting system needs to send the ASCII representation for the number 1, which is the ASCII value of 49, not the decimal value 1. To display the decimal value 127, the transmitting system would have to send the ASCII representations for the decimal numbers 1, 2 and 7 of the ASCII values of 49, 50 and 55. To capture the *SatPC32* commands as displayed above, I actually authored a PIC program dedicated to the translation of decimal values into ASCII that are in turn sent to *HyperTerminal*.

Talking Through the Software

The thought process behind the PIC software that is loaded into the microcontroller of the interface project goes something like this:

- Determine the operating mode desired from the setting of the mode switch of the interface, either U/v (UHF uplink/VHF downlink) or V/u (VHF uplink/UHF downlink).
- Wait for and read valid frequency commands from *SatPC32*.
- Store the frequencies into temporary variables for VHF frequency and UHF frequency as appropriate.
- Check the status of the PTT switch on the interface board.
- If the PTT switch is open (in receive mode), send the appropriate receive frequency to the

Table 4
***SatPC32* CAT Command Set with FT-817 Radio for VO52 Satellites**

Command	Function
0000000000	CMD LOCK ON
000000008A	CTCSS?
1458992001	RX FREQUENCY
4352529401	TX FREQUENCY
0152529407	OPERATING MODE

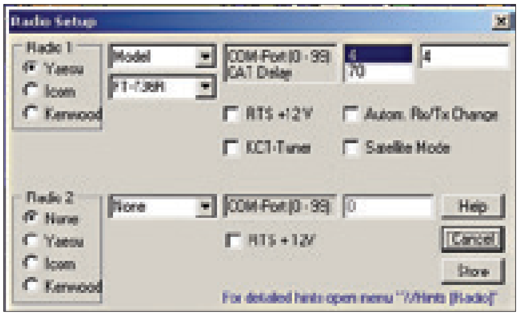

Figure 3 — *SatPC32* radio setup menu.

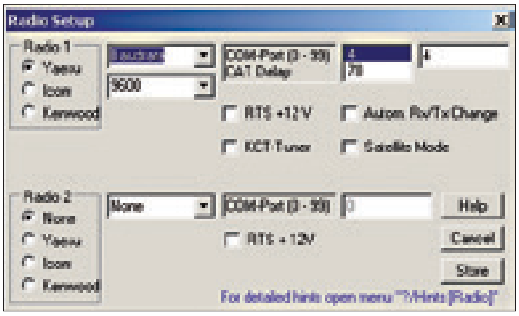

Figure 4 — Select the 9600 baud rate as illustrated.

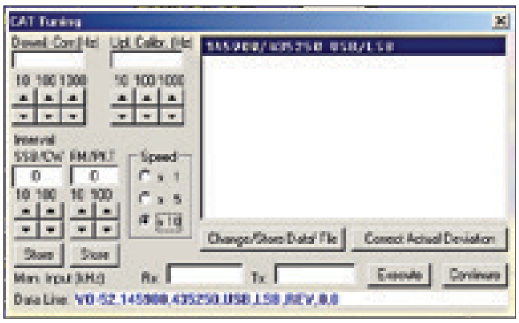

Figure 5 — Click on CAT in the menu bar and set the interval frequency to 0 for both SSB/CW and FM. Also select X10 speed as shown.

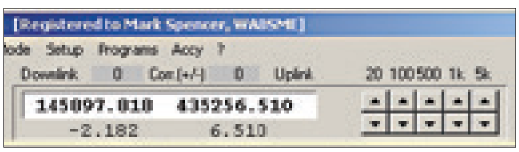

Figure 6 — Tune the radio frequency with the up and down arrows on *SatPC32*, not with the main tuning knob of the transceiver as shown.

FT-817 (depends on the MODE switch).

- a(1) Go back to step 1.
- If the PTT switch is closed (radio switched to transmit), send a command to toggle the VFO and then send the appropriate transmit frequency to the FT-817.
- Send a command to cause the FT-817 to begin transmitting.
- Hold the transmit mode until the PTT button is released.
- Go back to the first step.

This sequence may seem complicated, but if you take a moment to think about how you operate your radio manually, these are exactly the steps that you take. If programming a computer to do the work for you, you are just translating the things that you would do into commands that you want the computer to mimic.

Interface Connection

The interface draws power from the FT-817 ACC jack. This current source is always on and is not controlled by the rig's power switch. Because there is no ON/OFF switch for the interface, it is on and drawing current whenever it is plugged into the FT-817. Be aware of this and don't leave the interface plugged into the radio when not in use or you'll deplete the radio batteries. Use a standard USB to serial converter to connect the interface to the computer USB port. Set the FT-817 VFOA (RX) and VFOB (TX) to the correct band and modes for the satellite you are going to use. Set the correct operating mode for the satellite of interest, either V/u or U/v.

SatPC32 Setup

Use the Yaesu FT-736 radio in the Radio Setup menu of *SatPC32* as illustrated in Figure 3. Select the appropriate serial port for your computer setup. Select the 9600 baud rate as illustrated in Figure 4. Make sure that you also set the CAT baud rate of the FT-817 to 9600 baud. Click on CAT in the menu bar and set the interval frequency to 0 for both SSB/CW and FM. Store this setting for later use. Also, select X10 Speed (Figure 5). This setting makes the interface much more responsive during operation. Unfortunately this speed setting is not saved and defaults to X1 Speed each time *SatPC32* is launched, so you'll have to reset this each time you start the program. If you forget, the interface will act sluggish, but it will operate. When you want *SatPC32* to control your FT-817, click on the C icon in the upper left of the *SatPC32* menu bar.

Radio Operation

The *SatPC32* should start updating the frequency for the selected satellite. The data LED will flash in step with the frequency updates. You tune the radio frequency with the up and down arrows on *SatPC32*, not with the main tuning knob of the transceiver (see Figure 6). This may take a little getting used to. Start with the higher Hertz intervals for gross tuning and transition to the lower hertz tuning for fine tuning. Having up-to-date Keplerian data is more critical when using the interface to control the radio since you will not be able to hear your downlink to compensate for any Keplarian data erosion or computer clock inaccuracies.

With the radio tuned in to the desired receive frequency with the *SatPC32* tuning arrows, the program will calculate the proper uplink frequency. Once you have the receive frequency set, use the FT-817 RIT adjustment to make minor adjustments of the receiver frequency, to clarify the signal and compensate for minor adjustments for Doppler shift.

When you want to transmit, press the PTT switch on the interface board, not the PTT on the rig microphone (the interface does not read the status of the rig microphone PTT switch). There is a slight delay for the PIC to toggle the VFO and set the transmit frequency before the rig is put into transmit mode. Because the FT-817 does not allow for CAT frequency adjustments while the rig is in transmit mode, keep your transmissions short (this normally isn't a problem for satellite contacts). When done transmitting, simply release the PTT switch on the interface board to return to receive. The *SatPC32*/interface combination will return the rig to receive on the proper VFO and proper Doppler adjusted receive frequency.

Conclusion

Ideally, to operate the analog satellites you would want a full duplex radio that would allow you to monitor your downlink frequency while transmitting to permit you to track for Doppler shift. The interface described here, while not the ideal solution to operating a non-full duplex radio, can help you operate the analog birds with a non-full duplex radio, in this case the FT-817 being controlled by *SatPC32*. The interface concept serves as a model that can be adapted to other radios and satellite tracking software. CU on the birds.

Mark Spencer, WA8SME, is the ARRL Education and Technology Program Director. You can reach him at **mspencer@arrl.org.**

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