

The AMSAT[®] Journal

Editor-in-Chief
Joe Kornowski, KB6IGK

Assistant Editors
Bernhard Jatzeck, VA6BMJ
Douglas Quagliana, KA2UPW/5
Paul Graveline, K1YUB

Volume 43, Number 3

May/June 2020

000231 Tx 1780 + WA6DNR N7ZO CN85	
000252 10 0.2 1884 + N7ZO WA6DNR +08	FT4 Goes to Space!
000300 Tx 1780 + WA6DNR N7ZO R+10	
000307 7 0.1 1885 + N7ZO WA6DNR RR73	
000315 Tx 1780 + WA6DNR N7ZO 73	
000245 10 0.0 1239 + CQ W5SAT DM26	
000326 Tx 1780 + W5SAT N7ZO CN85	
000330 10 0.0 1237 + CQ W5SAT DM26	
000337 Tx 1780 + W5SAT N7ZO CN85	
000345 8 0.0 1239 + W5RKN W5SAT DM26	
000352 Tx 1780 + W5SAT N7ZO CN85	
000400 7 0.0 1239 + W5RKN W5SAT DM26	
000407 Tx 1780 + W5SAT N7ZO CN85	
000415 8 0.0 1238 + N7ZO W5SAT +14	
000422 Tx 1780 + W5SAT N7ZO R+08	
000430 5 0.0 1237 + N7ZO W5SAT RR73	
000437 Tx 1780 + W5SAT N7ZO 73	
000445 9 0.0 1238 + W5RKN W5SAT DM26	
000452 Tx 1780 + CQ N7ZO CN85	
000500 5 0.0 1239 + W5RKN W5SAT DM26	

FT4 Goes to Space!

In this issue —

Apogee View..... 3

User Services Update 4
by Robert Bankston • KE4AL

AMSAT's New Member and Event Portal 5
by Robert Bankston • KE4AL

Treasurer's Report 5
by Robert Bankston • KE4AL

Educational Relations Update 6
by Alan Johnston • KU2Y

The Recovery of LES-5 14
by Scott Tilley • VE7TIL

Operating WSJT-X FT4 on Satellites 19
by David Beumer • W0DHB;
Alan Bowker • WA6DNR;
Carlos Cardon • W7QL; Ronald G. Parsons • W5RKN; Brad Schumacher • W5SAT; Robert Wright • N7ZO

For Beginners — Amateur Radio Satellite Primer V..... 21
by Keith Baker • KB1SF/VE2KSF

Using the Kenwood TM-710 GA's Internal TNC with the FalconSat-3 BBS 24
by Kevin Zari • KK4YEL

Comparing Omnidirectional VHF Antenna Construction and Performance for LEO Satellites..... 26
by Gerard Anandappa • KD9NNU;
Abdus Saboor Zubair • KD9NNV;
Dan White, Ph.D. • AD0CQ

Regulatory Update 30
by Paul Stoetzer • N8HM



Periodicals
POSTAGE PAID
At Kensington, MD
and at additional
mailing offices

AMSAT
10605 Concord St., Suite 304
Kensington, MD 20895-2526

AMSAT CubeSatSim Activities with Fox Telem

Alan Johnston, Ph.D, KU2Y
Vice President, Educational
Relations, AMSAT
Associate Teaching Professor,
Villanova University
ku2y@amsat.org

Pat Kilroy, N8PK
Flight Systems Integration &
Test (I&T) Engineer
NASA Goddard Space Flight
Center
n8pk@amsat.org

Jim McLaughlin, KI6ZUM
Zum Radio
STEM Advocate
ki6zum@gmail.com

David White, WD6DRI
Professional Geologist
Lifetime Amateur Radio
Enthusiast
STEM Advocate
wd6dri@gmail.com

Introduction

In the previous issue of the *AMSAT Journal* (March/April 2020), we introduced new design updates to the AMSAT CubeSatSim

as an improved tool for education and demonstrations. In this article, we will describe several educational activities that can be enjoyed with the CubeSatSim using FoxTelem, the free telemetry analysis software used by AMSAT members for on-orbit CubeSats. Some of these activities were also applied to the original design of the CubeSat Simulator in the Jan/Feb 2019 issue of the *AMSAT Journal* (<http://cubesatsim.org/content/CubeSatSimPaper2.pdf>) based on the educational activities proposed for the original ARRL ETP CubeSat Simulator by Mark Spencer, WA8SME, in the *AMSAT Journal*, November/December 2009 issue (<http://www.arrl.org/files/file/ETP/CubeSat/CubeSat-Pt2-NovDec09.pdf>).

First, we will describe how to download and configure the FoxTelem software for the CubeSatSim. For those without an AMSAT CubeSatSim model at hand, we posted several WAV files that can be downloaded to get experience with real-time analysis of telemetry from the CubeSatSim. Next, we will discuss various activities that can be done and describe how to proceed step by step using FoxTelem.

Background

The AMSAT CubeSatSim, our CubeSat Simulator, shown in Figures 1, 2, and 3, is a Raspberry Pi Zero-based, 3D-printed, functional model of a 1U form factor CubeSat. The simulator is designed to act, as

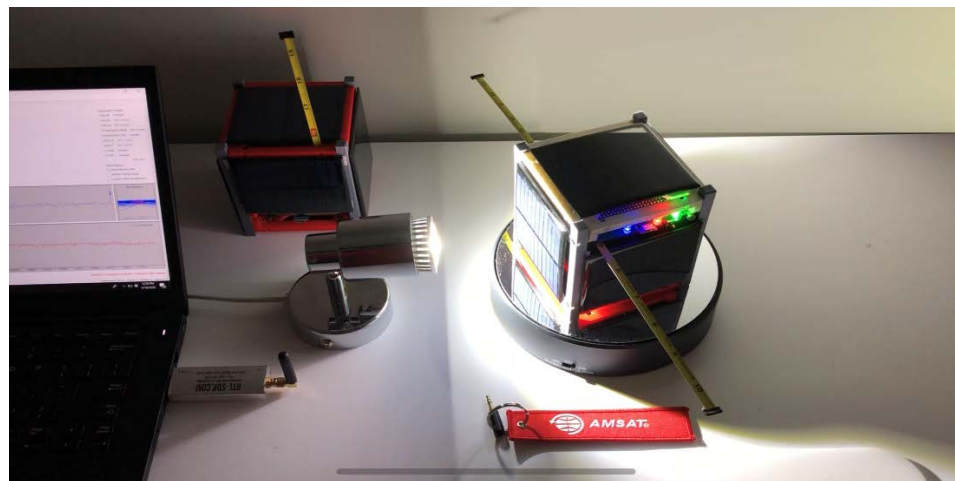


Figure 1 — The AMSAT CubeSatSim is shown on a rotating turntable in front of a sun simulator LED lamp, with FoxTelem running on a laptop and an RTL-SDR USB dongle serving as the ground station receiver. The tape measure monopole antenna is

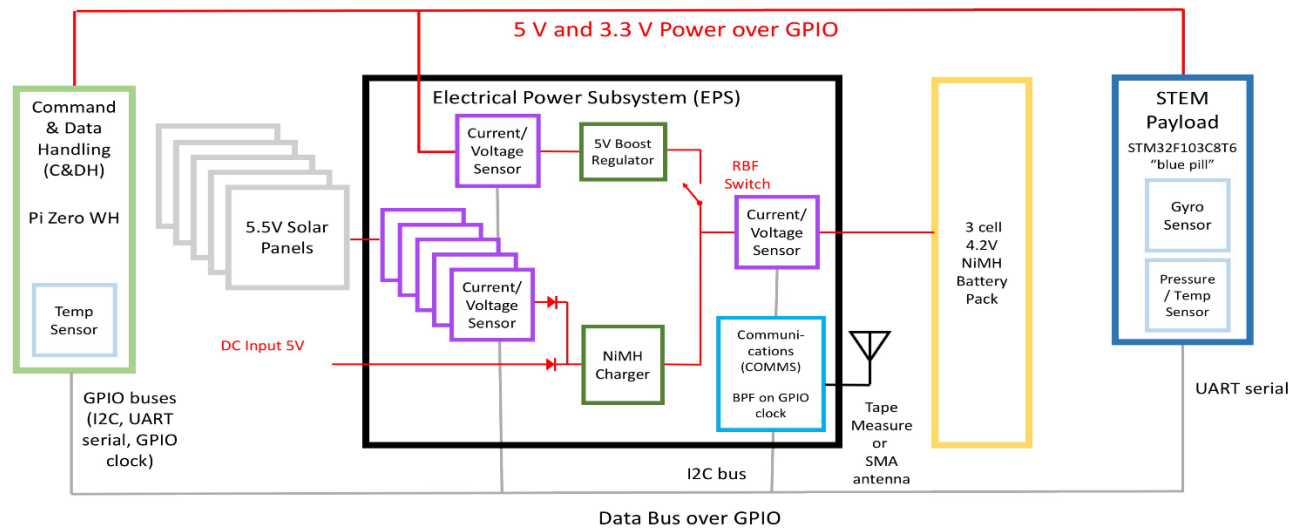


Figure 2 — The CubeSatSim Block Diagram showing the main functional blocks: Command & Data Handling (CD&H), Electrical Power Subsystem (EPS), Communications (COMMS) subsystem, Battery pack, and expandable STEM Payload.



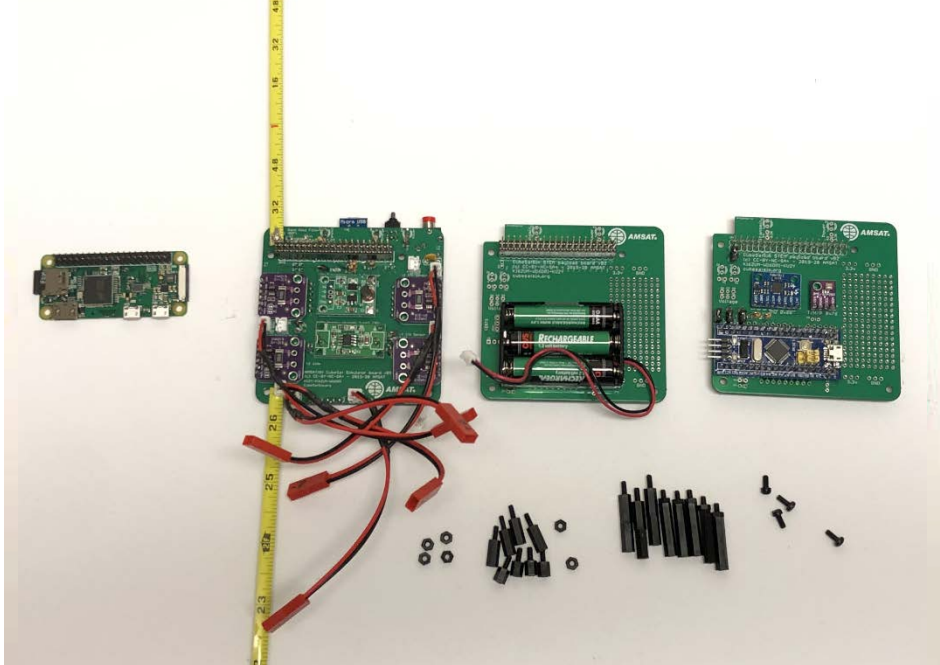


Figure 3 — The CubeSatSim board stack showing (from left to right) Raspberry Pi Zero WH (C&DH), Main Board (EPS and COMMS), Battery Board and the STEM Payload Board.

reasonably as possible, like a real one flying in Low Earth Orbit (LEO) to demystify how satellites work. The construction plans and software are fully open-sourced, and information about building one is available at cubesatsim.org. Like real LEO satellites, this simulator is self-powered through onboard rechargeable batteries and solar panels. It transmits telemetry on UHF. For details on the design and construction of the simulator, see our article in the March/April 2020 issue of the *AMSAT Journal* (<http://www.amsat.org/wordpress/wp-content/uploads/2020/04/The-AMSAT-Journal-March-April-2020.pdf>).

Getting Telemetry Using FoxTelem

FoxTelem is the open-source AMSAT Telemetry Analysis Tool by Chris Thompson, G0KLA/AC2CZ, that AMSAT uses for its on-orbit CubeSats such as Fox-1B, Fox-1Cliff, Fox-1D, HuskySat-1, and in the future Fox-1E. Many *Journal* readers already use FoxTelem. Even if you don't yet have an AMSAT CubeSatSim, we would encourage you to install FoxTelem and upload the telemetry you copied and decoded from a local overhead pass to AMSAT's servers. Your data is then displayed at <http://amsat.org/tlm> to support and share with all.

The CubeSatSim can operate in Fox-1 emulation mode, transmitting telemetry at 434.900 MHz in a format compatible with the FoxTelem software, allowing users to decode and graph telemetry in real-time. The ground station receiver at the input to FoxTelem can be as simple as the low-cost RTL-SDR USB dongle, the

audio output from an SDR application, or simply a saved WAV or a downloaded CubeSatSim example test WAV file (link below). The following steps will help you install and configure FoxTelem. FoxTelem is written in Java, which means that it can

run on your choice of Windows or MacOS, or Linux (as on the Raspberry Pi). For the latest installation instructions, see the CubeSatSim Wiki Ground Station page at <https://github.com/alanjohnston/CubeSatSim/wiki/Ground-Station>.

Support for the CubeSatSim is included in FoxTelem version 1.09 and later. The latest version for your operating system is available for download at <https://amsat.us/foxtelem/>. If version 1.09 or later is available, download it. If not, then you need to download a test version of 1.09 available at <http://www.g0kla.com/foxtelem/downloads/test/>

If you already have FoxTelem 1.09 or later installed, but you do not see CubeSatSim spacecraft tabs, then you need to add them. Under the menu Spacecraft, select Add and then select the file CubeSat_Simulator_DUV_fm.MASTER, and select Add. After a minute, the CubeSatSim-FSK tab will appear. Select Spacecraft Add again selecting CubeSat_Simulator_PSK_fm.MASTER to add the CubeSatSim-BPSK tab.

Once downloaded, you will need to extract all the files and follow the install steps guided by your operating system. When you run

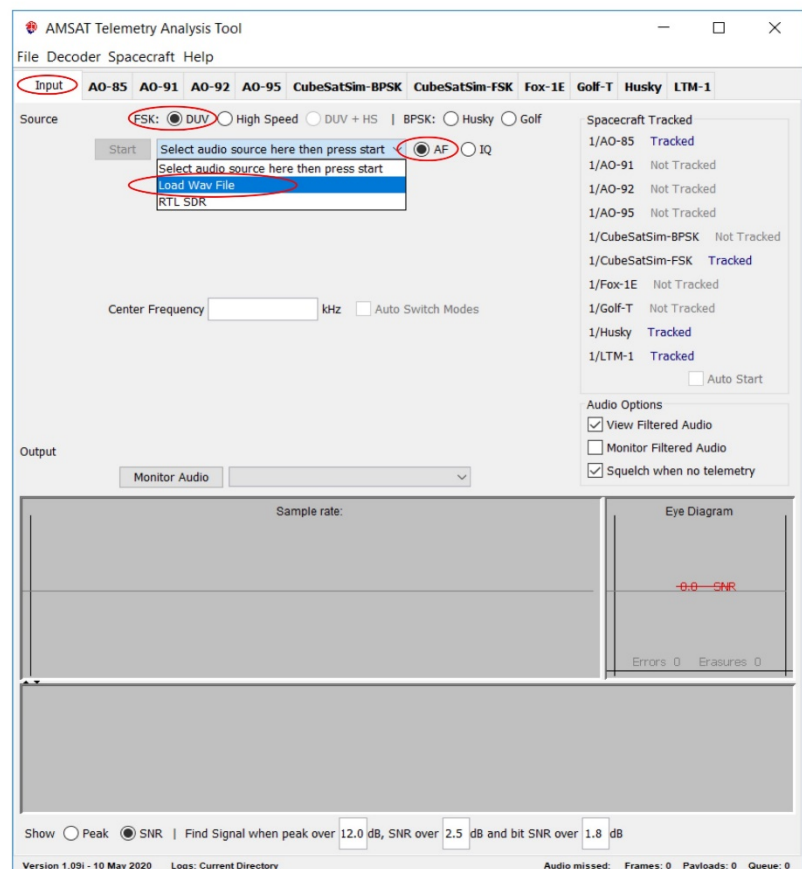


Figure 4 — AMSAT FoxTelem Telemetry Analysis Tool loading a test WAV file.



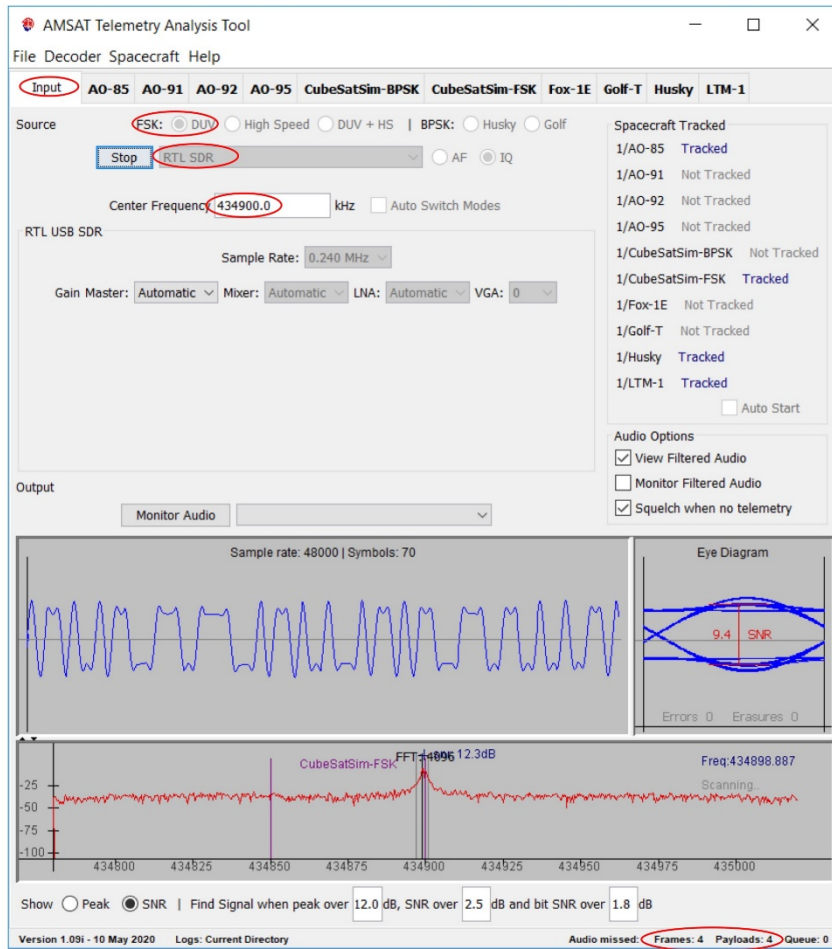


Figure 5 — FoxTelem with RTL-SDR input receiving CubeSatSim FSK telemetry.

FoxTelem, you may be prompted to update Java on your computer. If you have never installed FoxTelem before, you will get a “Welcome to the Amsat Fox Telemetry Analysis Tool” pop-up. Select Continue to choose the default location for your directory. You will then need to select “Yes” a number of times to install the spacecraft files. In a minute, the FoxTelem application will load.

You should see a CubeSatSim-FSK tab and a CubeSatSim-BPSK tab because we use different Fox IDs for the two different telemetry modes: Fox ID of 7 for the 200 bps FSK (Frequency Shift Keying) mode that emulates Fox-1B, Fox-1Cliff, and Fox-1D telemetry, and Fox ID of 99 for the 1200 bps BPSK (Binary Phase Shift Keying) mode that emulates HuskySat-1 and Fox-1E telemetry.

You can test your FoxTelem installation by downloading the test WAV files from cubesatsim.org/wav. Here’s how to use them: in the Input tab, under FSK select the DUV

option. On the audio source, select the AF option. Next, click on the “Select audio

source here then press start” drop-down menu, and select “Load Wav File” as shown in Figure 4.

A file menu will open, and you should select the WAV file you downloaded in the previous paragraph, then click on Open and then the Start button. You will see the data being processed. The Frames and Payloads counts at the bottom of the window should increase. This means you are ready to go!

If you have an RTL-SDR USB dongle plugged in, you can receive live telemetry from your CubeSatSim if it is transmitting FSK telemetry. In the Input tab, make sure you still have selected, under FSK, the DUV option. In the Source drop-down menu, select RTL SDR. Set the frequency by clicking on the Center Frequency box and typing 434900 and then click Start. Drag the bottom of the FoxTelem window down, and the FFT spectrum plot window will appear. If your CubeSatSim is transmitting nearby, you should be receiving the signal, as shown in Figure 5.

If no signal appears from your CubeSatSim, you instead may see something similar to what appears in Figure 6 in FoxTelem. Note there is no peak in the FFT spectrum at the bottom. You can ignore the warning at the bottom. Having both Find Signal and Doppler Tracking disabled is not a problem

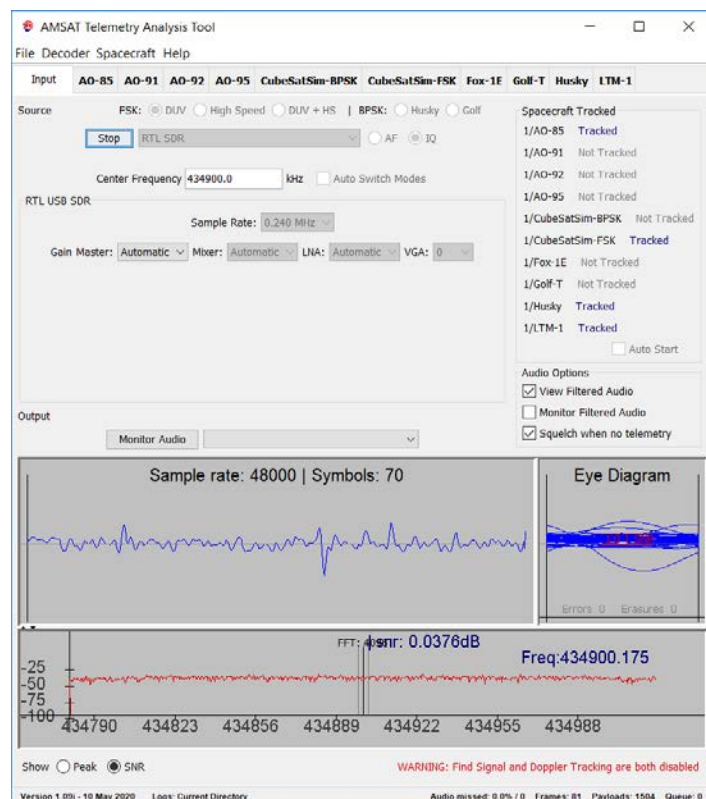


Figure 6 — FoxTelem with no signal from CubeSatSim.



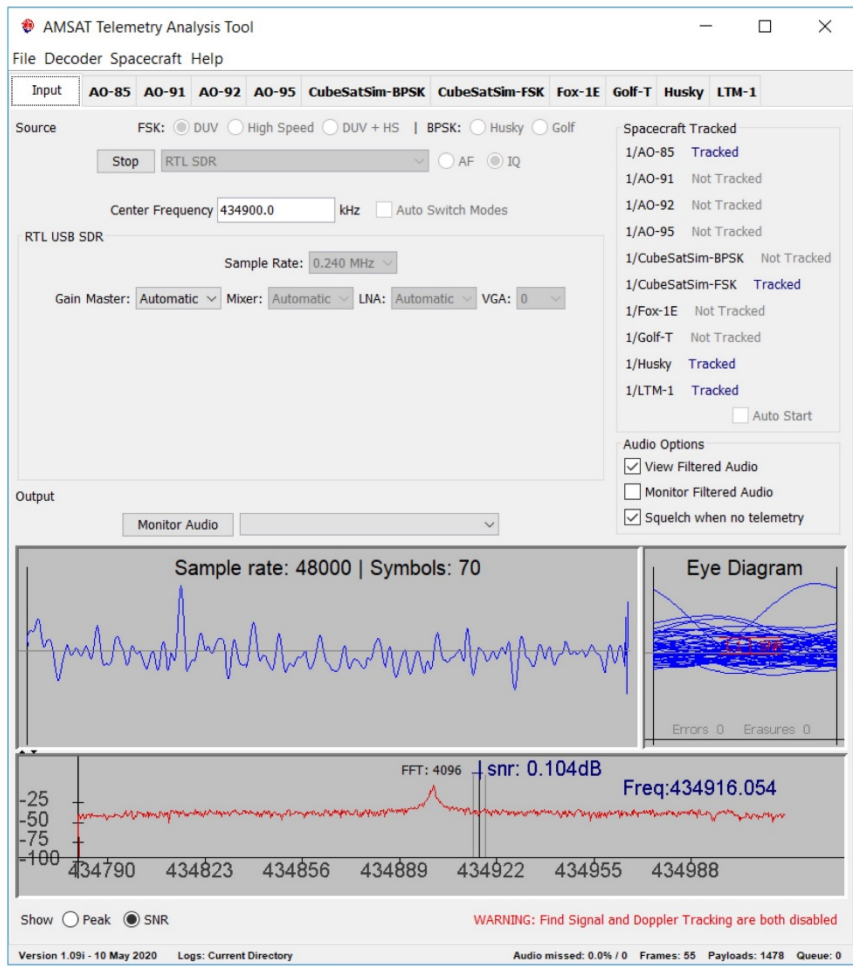


Figure 7 — FoxTelem with signal from CubeSatSim but not tuned to signal.

for the CubeSatSim as they can be enabled under File/Settings.

You may also see a signal from the CubeSatSim but without the Frames or Payloads count increasing. This might be because FoxTelem is not tuned to the signal, as shown in Figure 7.

To get FoxTelem to tune the signal, click on the peak in the FFT window at the bottom, and you should start to see telemetry decodes. The Frames and Payload counts, as seen at the bottom of Figure 5, should increase every four seconds or so.

When you click on the CubeSatSim-FSK tab, you will see the spacecraft Health telemetry data, sometimes also known as “housekeeping telemetry,” as shown in Figure 8. There should be data under the RT (“Real-Time” values) column. The MIN and MAX columns will show zeros as the minimum and maximum value functionality has not yet been implemented in the CubeSatSim. Uncheck the Display Raw Values checkbox at the bottom if it is checked.

If all values show as zeros, that means that you have not successfully decoded any telemetry frames. Note that the modulation transmitted by the CubeSatSim (FSK) must be the same as the Source setting in the Input tab (DUV FSK), and the satellite tab (CubeSatSim-FSK). If any of these are incorrect, you will not see data.

Note, too, that if the CubeSatSim is run in BPSK telemetry mode, FoxTelem can be set in BPSK mode as well, and similar telemetry will decode in the CubeSatSim-BPSK spacecraft tab. The fields are essentially the same, although some have slightly different names.

Assuming you have made it this far, congratulations! You are now ready to gather the telemetry data in this article. To see a real-time graph of any parameter, double click on the value or parameter name in Figure 8, and a new window will open with a real-time graph displayed.

The remainder of this article will describe a set of activities in the form of questions that can be done in a classroom or in demos using the CubeSatSim and FoxTelem to explain various aspects of radio, communications, orbital space, and satellites. Some of these educational activities were proposed in the original ARRL ETP CubeSat Simulator

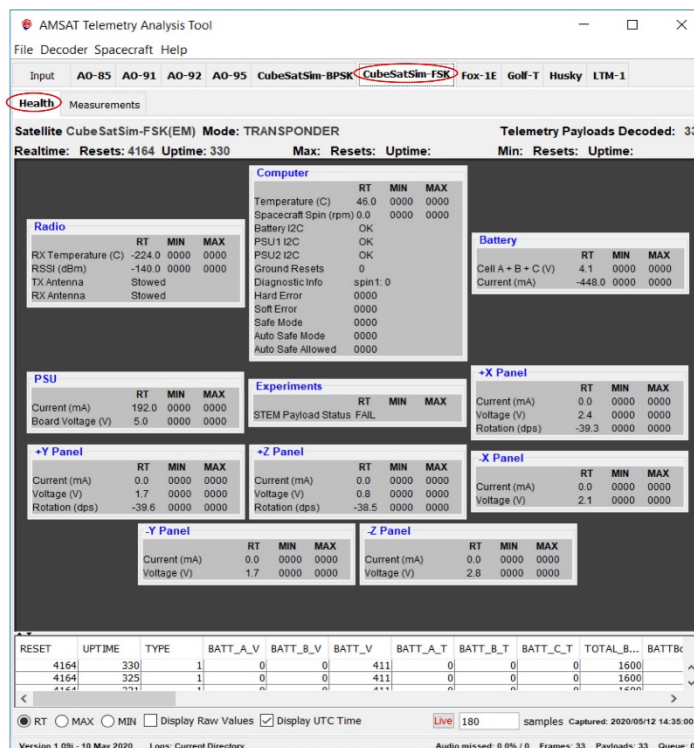


Figure 8 — FoxTelem CubeSatSim-FSK Data, Health Tab.

materials by Mark Spencer, WA8SME, in the *AMSAT Journal* November/December 2009 issue (<http://www.arrl.org/files/file/ETP/CubeSat/CubeSat-Pt2-NovDec09.pdf>).

Radio and Data Communication

The CubeSatSim uses digital transmission of telemetry data. Data about the health and environment of the CubeSatSim is gathered by the Raspberry Pi in the C&DH using onboard sensors. This includes voltages, currents, and temperatures. This information is then grouped into a "payload" packet of data which is then encoded and modulated as a "frame" and sent over the radio channel as RF (Radio Frequency) signals, and down to the ground station — your computer with FoxTelem and the RTL-SDR USB dongle receiver.

Can you tell the signal received at your ground station is binary data?

If you look at the Sample window in FoxTelem when data is being received and decoded, such as in Figure 5, you can see the binary nature of the data. The waveform looks like a square wave, with the amplitude having only two states: high or low.

Do you have an open eye pattern or a closed eye pattern?

If you look at the Eye Diagram window next to the Sample window in FoxTelem when data is being received and decoded, such as in Figure 5, you will see what communication engineers call an "eye pattern." The eye pattern is generated by superimposing the binary waveform from the Sample window so that it is just one bit wide and repeats over and over again. If the binary waveform is only assuming the high and low values, the lines in the eye pattern will be mainly at the top and bottom, with some going up or down on the sides when the waveform is transitioning from high to low or low to high. If the area in the middle is empty, this is described as an "open eye pattern." Figure 6 shows a "closed eye pattern," where the waveform is just random noise. The eye pattern gives a quick visual view of the quality of the received signal and is a good indication of whether the telemetry signal will successfully be demodulated and decoded. Note that because the CubeSatSim is typically right next to the ground station, this eye pattern is amazingly clear. A real satellite signal such as one from one of the Fox-1 CubeSats will not look quite as perfect.

What is the actual transmit frequency of your CubeSatSim?

If you look at the FFT window at the bottom of FoxTelem, you will see a section of the radio spectrum centered about the Center Frequency. In Figure 5, the Center Frequency is 434.900 MHz, and the FFT shows signals from 434.780 to 435.020 MHz (note that this corresponds to the Sample Rate setting of 0.240 MHz for the RTL-SDR). When FoxTelem is tuned to a signal, it shows the frequency next to Freq. In Figure 5, the actual frequency according to FoxTelem is 434898 kHz or 434.898 MHz.

What does the telemetry signal sound like if you listen to it?

A UHF radio tuned to 434.900 MHz will allow you to listen to the transmitted signal. Instead, if you click on the Monitor Audio button, you can hear the audio of this digital signal. It sounds like a low pitched rumbling sound; some describe it as a "background freight train." The Fox-1 satellites send this when the FM transponder is active in the background of the voices making contacts. You might be able to hear it on the air, although most radios filter out the low frequencies making it difficult to hear. To hear it in the Windows application SDR#, you need to uncheck the Filter Audio option under the Audio settings.

Can you hear the Morse Code CW callsign transmitted by your CubeSatSim?

When the CubeSatSim first turns on, it sends an FM modulated CW callsign. If you watch the FFT window during the CubeSatSim startup, you will see the individual tones being sent. (Using the pushbutton on the CubeSatSim, you can turn on, reboot, or shut down the CubeSatSim.) After a short break, the constant FSK telemetry will start. If you uncheck the Squelch When No Audio button, you can hear the CW tones, but turn down your volume first — loud un-squelched FM can be very annoying!

What is the Signal to Noise Ratio (SNR) of the received telemetry signal?

When the telemetry signal is being decoded, the FFT will indicate the Signal to Noise Ratio (SNR). Figure 5 shows this as 12.8 dB (decibels or dB are a logarithmic scale useful for audio and radio signal intensity levels). The higher this number, the stronger the level of the signal (the peak) over the background noise level (the flat line across the FFT). Another SNR appears in the Eye Diagram window. In Figure 5, this shows as 9.4 dB. This indicates how "open" the eye is. A higher number indicates a cleaner signal with a better chance of decoding.

Basic Power and Electronics

What are the battery voltages and currents used or produced by the CubeSatSim?

The basic power subsystem information on the CubeSatSim is available in the CubeSatSim-FSK tab in the Health tab, after telemetry frames have been received and decoded, as shown in Figure 9. Examine the following fields: under Battery, Cell A + B + C (V) gives the voltage (3.9 V) of the battery pack, the three NiMH AAA cells in series. When you plug in the micro USB power supply to charge, the battery pack voltage will jump to over 4.0 V. Figure 9 also shows that the battery is supplying a current of 307 mA. A battery voltage of 3.9 V produces 1.2 W of power. Over a long run, the battery pack voltage averages 3.6 V and the battery current 246 mA. This gives an average power of 886 mW.

When you plug in an external power source for charging, you will see a "negative current" while current flows into the battery to store energy, such as in Figure 8. You may see more than 300 mA of charging current for a battery with a great depth of discharge, dropping to around 90 mA when the battery is nearly fully charged. Charging time is just over two hours for the AAA battery pack.

What are the Power Supply voltage and current of your CubeSatSim?

As shown in Figure 9, under PSU (Power Supply), the Current (mA) gives the Electrical Power Subsystem (EPS) bus current (192 mA), and the Board Voltage (V) gives the EPS bus voltage (5.0 V).

In the CubeSatSim EPS, an electronics circuit known as a Boost Regulator (see Figure 2) turns the battery voltage, varying between 4.2 and 3.2 V, into a constant 5 V needed by the CubeSatSim. You will almost always see 5.0 V for the PSU Board Voltage in your telemetry.

The PSU Current averages about 156 mA over long runs and is about 44 mA less than the original CubeSatSim design. Since the new design also features the new STEM Payload Board, which draws about 35 mA, the real current reduction from the previous design is about 80 mA. The power is about 780 mW.

The efficiency of the CubeSatSim power conversion is approximately $780 / 886 \times 100 = 88\%$.



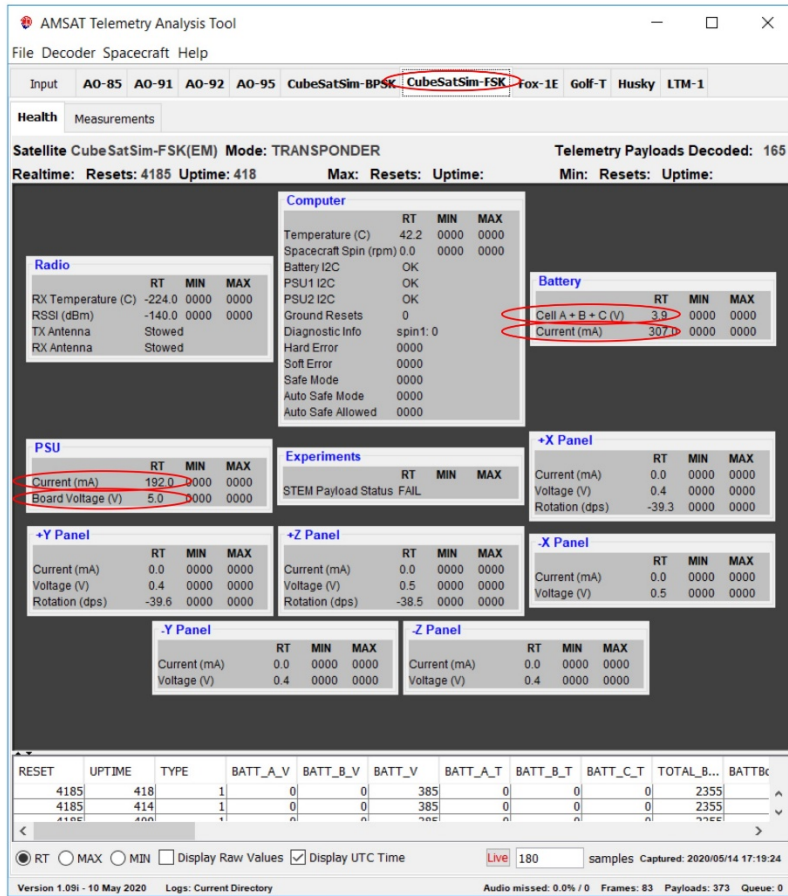


Figure 9 — FoxTelem CubeSatSim-FSK battery and power supply telemetry data.

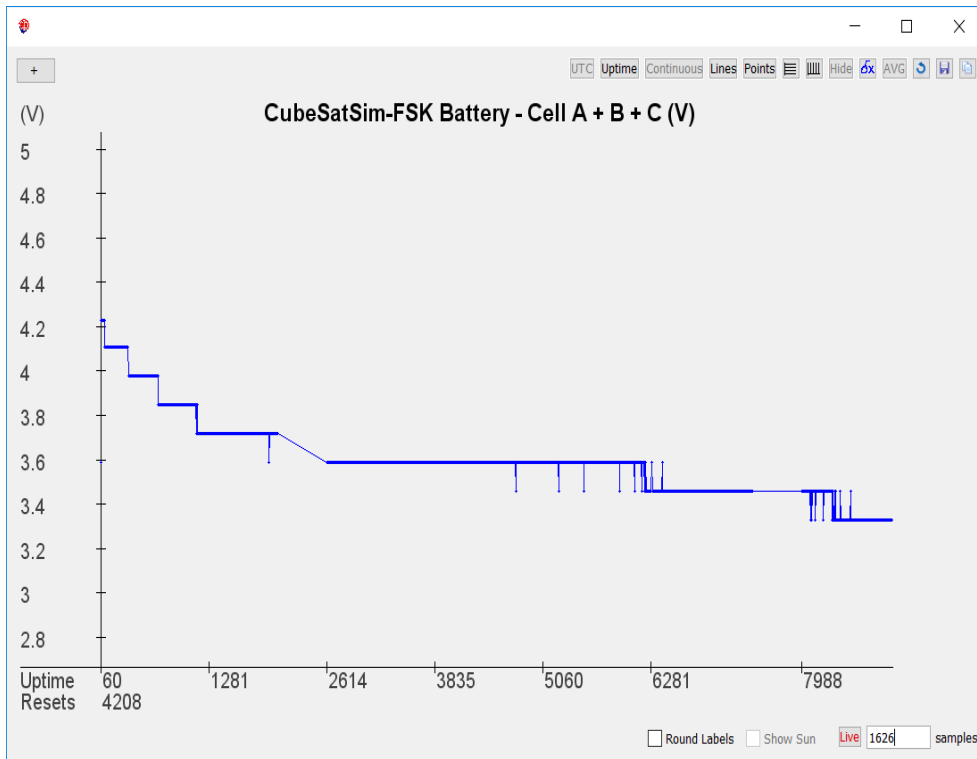


Figure 10 — FoxTelem CubeSatSim-FSK battery and power supply telemetry data.

What is the minimum panel/battery voltage required for the CubeSat Simulator to operate?

While the battery energy depletes, you will discover that the CubeSatSim safely and automatically shuts down (enters its “Safe Mode”) when the battery reaches a certain voltage. That minimum voltage is 3.2 V, although the last telemetry frame you see might have a slightly higher voltage.

Satellite Operation Activities

How long will the satellite operate in eclipse?

In eclipse (darkness), the CubeSatSim runs only on battery power. The test run of Figure 10 shows the battery pack voltage versus time until the CubeSatSim automatically shuts down. Over this period, the three AAA cell NiMH battery pack goes from 4.2 V down to 3.2 V. The 3.2 V is the lower limit since it is not recommended to discharge a NiMH cell below 1.1 to 1.0 V. The software shuts down the CubeSatSim safely at that point to protect the integrity of the file system on the Pi’s microSD memory card. The time axis of Figure 10 shows Uptime, which is the number of seconds since the Pi powered up. The battery discharge took 9005 seconds or about 150 minutes. For a typical LEO CubeSat, this is more than a complete orbit, which would be a good margin.

The NiMH AAA batteries are rated at 600 mAh in capacity each, which means the three have a stored energy of 600 mAh x (0.001 A/mA) x (3 batt/pack) x (1.2 V/batt) x (3600 sec/h) = 7800 Joules. During the CubeSatSim eclipse run, we had an average battery voltage of 3.6 V and an average current of 245 mA over 150 minutes, which is about 7900 Joules. This is very close to the rated stored energy.

How can you use the telemetry data to determine the rotation rate of the satellite?

The rotation rate can easily be determined by studying the graph of any one of the solar panels. If the CubeSatSim is nearly in darkness, it can be calculated from the solar panel voltage peaks. If rotating in sunlight or in front of a halogen lamp, it can be calculated from the solar panel’s current peaks. Figure 12 shows a solar panel voltage graph of the CubeSatSim rotating in front of an LED lamp. There is a separate plot of the voltages for each of the six solar panels, which are labeled using the spacecraft X, Y, and Z axes.

In three dimensional space, the spacecraft X, Y, Z axes are useful, since you can’t say

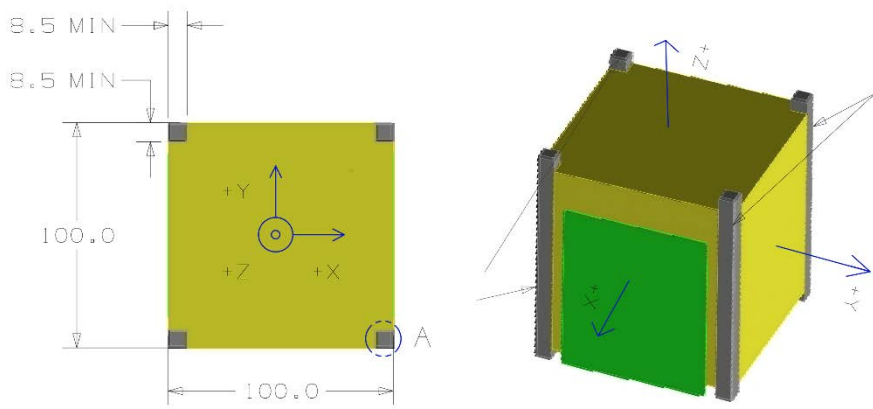


Figure 11 — Figures from the CubeSat Design Specification Standard document Rev 13 showing the +X, +Y and +Z axes on a CubeSat. Source: <http://www.cubesat.org/resources>

or the velocity vector, or the radiator on the +Y side to be facing deep space (or opposite the -Y solar array side), or the +Z to be Nadir pointing (earth facing side). Or a combination of the above, but your mileage may vary based on your mission requirements.

In closing the point of analyzing satellite rotation rate for a point on its orbital track using solar panel peaks, we assume that a peak occurs when the plane of the panel is normal to the source of the light, or closest to it. At the next peak for the same panel, one only needs to extract the timing between the two peaks to obtain the average rate in Rotations Per Minute (RPM). But be aware of potential ambiguities, such as the correct answer could be masked in an integer multiple of the RPM figure depending on one's sampling rate and telemetry reporting rate. Furthermore, inspecting the peak sequence in more than a single panel might reveal additional information (or confusion) on any tumbling rate or orientation. Fortunately, one need not worry about these higher-order challenges with your CubeSat simulator situated on a lab table and turntable. Unless, of course, your instructor mounted the simulator on the turntable in an uncharacteristic orientation, and without allowing a visual of it, challenged you to try to determine its rate and position, only by the telemetry at hand.

As you can see, depending on how precise your housekeeping sampling rate is, and the downlink frequency of the data of interest, rotation rate determination from solar panel performance is an excellent educational exercise. A flight model, however, might require a means a little more elegant and reliable. Note that when the Gyro on the STEM Payload Board is running, this could be seen directly in the Rotation field.

Conclusion

We have shown several activities that can be performed using FoxTelem to analyze CubeSatSim housekeeping telemetry. These are critical lessons to help you understand your satellite in orbit, and in the care and feeding of it, which will guarantee the longest lifetime possible. Developing skills

like these will make you more marketable and valuable in the aerospace industry throughout your career.

In a future article, we will focus on interfacing new sensors, experiments & payloads with the AMSAT CubeSatSim. 🌐

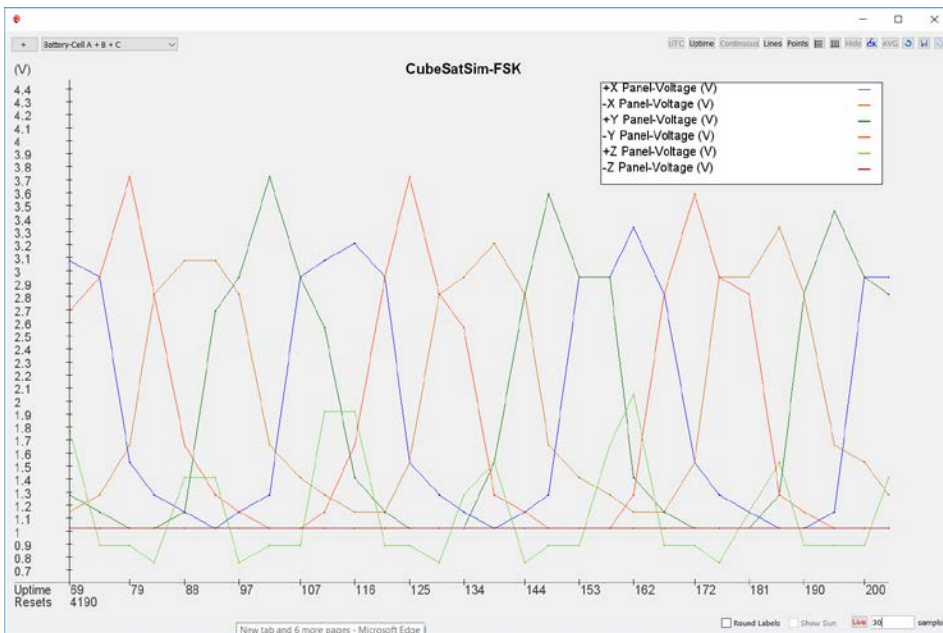


Figure 12—FoxTelem Graph of CubeSatSim solar panel voltages on the X, Y, and Z axes.

up or down very well. Figure 11 is from the CubeSat Design Specification Standard and shows the X, Y, and Z axes as labeled. The left figure shows a view looking down on the CubeSat, showing the X and Y axes as used for dimensional graphs. Typically, the +X axis points to the right on the page while the +Y axis points up on the page. The third dimension, +Z, can be thought of

as pointing out of the page, as given by the Right-Hand Rule. The -X, -Y, and -Z axes are in the respective opposite directions. The right figure shows these on a CubeSat space frame.

Furthermore, some spacecraft coordinate system's naming convention requires, for example, their +X to face into the "Ram"



How-to Options to Make Your Own

The opportunities to learn CubeSat design and fabrication abound for AMSAT members. From Maui to Maine, and from the tip to toe of North America, AMSAT members and university students possess a gold mine of demystifying satellite design details to start a design of their own. The stepping stones of the engineering processes, from a fully functional satellite simulator (Sim) to an Engineering Development Unit (EDU) to an actual Flight Model (FM), are at your fingertips with your personal AMSAT resources.

Example 1:

Run, don't walk, to the AMSAT Annual Meeting & Space Symposium Proceedings for six years in a row starting with Orlando (2012) and completing in Reno (2017). Substantial engineering documentation for the five Fox-1 flight models is laid out in the latter pages of each publication. You'll see Concept of Operations (ConOps), requirements specifications, Interface Control Drawings (ICDs), subsystem design specs and schematic diagrams, Flight Software (FSW) Architecture Specs, Mechanical Drawings, Bus Pin Assignments, experiment/payload details and so much more. It's the real deal!

Example 2:

The seeds of the low-cost, high-educational value AMSAT CubeSat Simulator (CubeSatSim) were sown at that very AMSAT Symposium in Reno. One year later, in Huntsville, we unveiled the AMSAT CubeSat Sim proof of concept model. Then last year, in Arlington, we demonstrated the first CubeSat Simulator for beta testers to build, and simultaneously released four completed & tested kits along with ground station components to the AMSAT public, shipping them all around the country for inspection, orientation, education and outreach demonstrations. In between the AMSAT space symposia, we published details of building and operating them in parts, via sequential issues of the AMSAT Journal. A repository of these issues and proceedings papers is found at CubeSatSim.org. Our CubeSat Simulator developments continue.

Example 3:

In the previous issue of the AMSAT Journal, March/April 2020, George Downey and Bob Bruninga, WB4APR, shared a seriously intriguing and totally amazing CubeSat flight model they built and tested.

the flight model to behold a "LabSat" for classroom and public demo use, not unlike our CubeSatSim. Bob released an ocean of additional technical details to his "PSAT-1U" model online at <http://aprs.org/psat1u.html>.

Epilog

An exercise left to the "student" is to inspect each of the three examples for the processor type used in each example above in the respective Internal Housekeeping Unit (IHU) — or Command & Data Handling (C&DH) subsystem as some call that part holding the central "brains" of a satellite. Because we engineers must function in the real world where it is the scientists who often create the purpose of a given satellite mission, we ought to consider how ready and capable the interface is to accommodate a scientist's payload experiment to the satellite bus. "The payload is the reason for the season." And the budget. A key to this accommodation is in the critical design of the C&DH.

