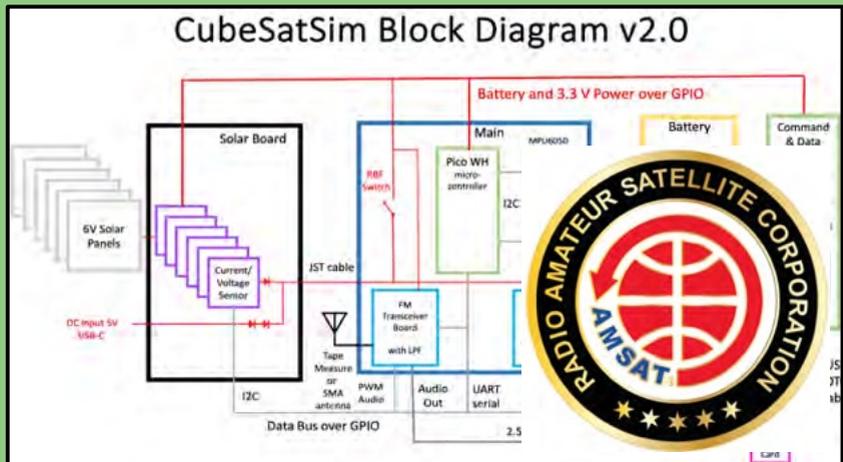
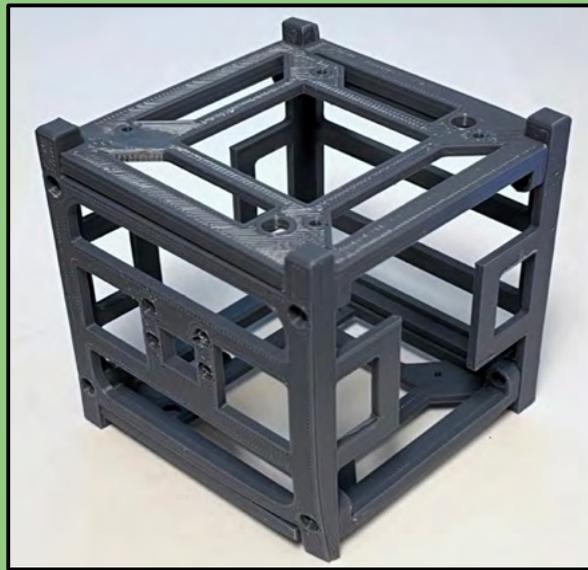
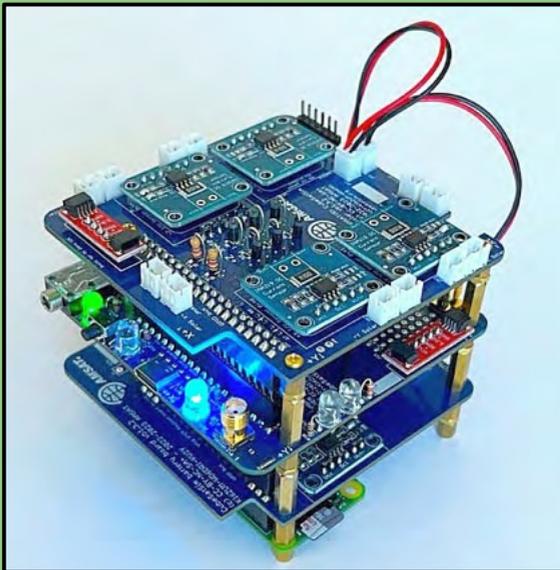


Educators Guide to Using Satellites in the Classroom



Blinks	Mode	Description	Decoding	Command	Audio	Waterfall(WebSDR)
1	APRS	Automatic Packet Reporting System. This digital mode sends a packet of data with AFSK or Audio Frequency Shift Keying modulation.	Windows: SoundModem or Diteware! Raspberry Pi/Linux: OpenWebRX or Diteware!	a	CubeSatSim.org/a	
2	FSK	Frequency Shift Keying. This mode transmits a continuous signal that emulates the AMSAT Fox CubeSats such as Fox-1C or AO-95. Also known as DUV or Data Under Voice.	Windows/Raspberry Pi/Linux: FoxTelem	f	CubeSatSim.org/f	
3	BPSK	Binary Phase Shift Keying. This mode transmits a continuous signal that emulates the AMSAT Fox-1E CubeSat. You demodulate using USB.	Windows/Raspberry Pi/Linux: FoxTelem	b	CubeSatSim.org/b	
4	SSTV	Slow Scan Television. This mode transmits stored images in Scottie 2 format which sounds like a series of tones.	Windows: MMSSTV Raspberry Pi/Linux: QSSTV	s	CubeSatSim.org/s	
5	CW	Continuous Wave or Morse Code. This mode transmits a FM modulated tone at 20 words per minute Morse Code telemetry.	Windows/Raspberry Pi/Linux: fDigi with spreadsheet http://cubesatsim.org/fel.html	m	CubeSatSim.org/m	
6	Cross Band Repeater	Simulates a U/V (UHF uplink, VHF downlink) FM satellite repeater. Default uplink is 435 Mhz and downlink is 144.9 Mhz.	OpenWebRX	e	CubeSatSim.org/e	
7	Fun Cube	FunCube simulates the BPSK telemetry of the FunCube series of CubeSats such as AO-73.	Raspberry Pi: FunCubeTelem	j	CubeSatSim.org/j	



Contents

Front Cover	1
Table of Contents	2
Preface	3
Introduction	4
AMSAT® CubeSatSim Project Wiki	15
CubeSatSim Quick Start Guide	24
Ground Station Quick Start Guide	26
Cross Band Repeater Quick Start Guide	28
Command and Control Quick Start Guide	30
CubeSatSim Activity Guides	32
AMSAT Youth Initiative Update	34
AMSAT Education and CubeSat Simulator Update	37
Bridging Orbit and Classroom: SatNOGS and CubeSatSim	56
New CubeSatSim Features and Capabilities	61
Building AMSAT CubeSatSims in the Classroom	72
CubeSatSim YouTube Playlist Links	100
AMSAT CubeSatSim Glossary	103
Radio Amateur Satellite Corp. Strategic Plan - 2021-2031	110
Join AMSAT	111

Preface

This guide makes extensive use of hyperlinks to Internet accessible references so the reader may explore the subject or topic referenced in the text. We have selected landing pages believed to be accurate and reliable (“evergreen” or long-lasting) to the best of our knowledge.

Any underlined, **boldface** and **blue text** is an Internet **hyperlink**. Depending on your PDF reader software settings (e.g. Edit > Preferences > Internet) you should be able to “left-click” on the hyperlink with your mouse and go directly to the web page. This guide was put together in early Fall, 2025. If you find an error or if you would like to discuss any aspect of this guide please contact:

Alan Johnston, Ph.D.
VP, Educational Relations
Radio Amateur Satellite Corp. (AMSAT)

- - -

Email: ku2y@arri.net

The Radio Amateur Satellite Corporation (AMSAT) was formed in the U.S. District of Columbia in 1969 as an educational organization. Its goal was and still is to foster Amateur Radio’s participation in space-based research, provide K-12 STEM education resources and further develop amateur satellite communication.

AMSAT was founded to continue the efforts, begun in 1961, by Project OSCAR, a west coast USA-based group which built and launched the first Amateur Radio satellite named OSCAR (Orbiting Satellite Carrying Amateur Radio) on Dec. 12, 1961.



Today, the “home-brew” flavor of these early Amateur Radio satellites lives on, as most of the hardware and software now flying on even the most advanced AMSAT satellites is still largely the product of volunteer effort and donated resources.

The AMSAT Education Initiative has expanded and is being modernized. Some of these changes are described on the pages of this Guide. Contact Professor Johnston to discuss any aspect of our Education Initiative.



Introduction: AMSAT Educators Guide

by David A. Vine
WA1EAW

Why Are Satellites Important Teaching Tools?

Space is more accessible to [everyone](#) today. Even K-12 classes assist in the construction of small Earth orbiting satellites. The cost of space launches is falling rapidly and fast becoming commonplace. Technology needed for space satellite operation has vastly improved performance, resulting in smaller size and lower cost.

The ***excitement and attention of students*** is easily captured and held when they participate in the conception, design, and other phases of small satellite construction. In fact, “ownership” of a science and research-oriented CubeSat is a prestigious symbol of achievement for colleges and universities around the world.

For your convenience, an extensive glossary of abbreviations, acronyms and terms can be found at the end of this Guide. All text shown in bold, blue and underlined type are active Internet hyperlinks to explanatory information.

Case Study

[University of Maine's](#) (UMaine, Maine, ME) [MESAT1](#) was launched into space onboard a [Firefly Aerospace](#) rocket on July 4th, 2024. MESAT1 carries three payloads designed by students in Falmouth High school, Fryeburg Academy, and Saco Middle School. UMaine and [USM engineering](#) teams collaborated on building the satellite and getting it ready for launch.

MESAT1 was developed to get students involved in space science. The ***cost-effective CubeSat*** is outfitted with four multi-spectral cameras designed to capture climate-related Earth imagery for experiments designed by the Maine K-12 students. It also includes an AMSAT provided LTM-1 linear transponder. AMSAT's Linear Transponder Module is a flight-proven, plug-and-play communications solution for CubeSat programs.

LTM-1 includes CubeSat command and control, spacecraft telemetry and a space science data downlink and services to [Amateur Radio](#) enthusiasts and others [around the world](#). LTM-1 enables satellite command and control, the ability to connect to a satellite and upload commands, settings, schedule operations and experiments, and perform other important functions.

At the request of University of Maine, AMSAT [designated](#) MESAT1 as OSCAR 122 (cataloged as MO-122). It's traditional for Amateur Radio satellites to carry the name [OSCAR](#), for "Orbiting Satellite Carrying Amateur Radio."

MESAT1 is a 3U [CubeSat](#) (meaning it is the size of three 1CubeSats combined, or 30 cm x 10 cm x 10 cm) assembled at UMaine. It is the ***first satellite of its kind ever built in Maine***. MESAT1 carries three imaging experiments proposed by Maine school students. AMSAT coordinated the [Radio Communication](#) capability for ground control and the Amateur Radio transponder.

"Access to space data motivates and excites students to learn fundamental skills in mathematics and science and helps them see the importance of STEM careers," said UMaine Vice President for Research and Dean of the Graduate School [Kody Varahramyan](#), who led the establishment of UMaine's [Space Initiative](#). "Being selected as a finalist for SmallSat Rookie of the Year is a national recognition of what Maine students and their faculty researchers can accomplish together."

Student Experiments

A statewide competition in 2019 drew payload proposals from schools across Maine. Three [student experiment projects](#) were selected for the MESAT1 mission: ALBEDO, IMAGER, and HAB.

ALBEDO: Saco Middle School will study reflected light (albedo) and local temperature in urban and rural areas, with the idea that urban heat islands could be mitigated through architectural designs that reflect more light.

IMAGER: Fryeburg Academy will photograph shallow coastal waters as part of an effort to distinguish turbidity and phytoplankton concentration from space. The academy is collaborating with the Wells National Estuarine Research Reserve.

HAB: Falmouth High School will work on early detection of harmful algal blooms by measuring atmospheric temperature and water vapor levels around bloom areas. They will watch blooms develop, move, and disperse.

During the 2025 [Small Satellite Conference](#) MESAT1/MO-122 was a finalist in the American Institute of Aeronautics and Astronautics ([AIAA](#)) [SmallSat award program](#) due to its mission of empowering K–12 students to contribute to the SmallSat award program. “***This recognition puts Maine on the map as a state that not only builds satellites, but also builds opportunities — for students, teachers, researchers and entrepreneurs,***” Varahramyan said.

Additional information about classroom activities and student projects is covered in much more detail later in this Guide.

Space Satellite Background Information

CubeSats are often launched into what is called Low Earth Orbit ([LEO](#)). LEO launches offer significant benefits to a CubeSat owner/operator and users due to its proximity to Earth. Low Earth Orbit altitude is 1,200 miles (2,000 km) or less. A LEO orbit around Earth usually takes 128 minutes or less, more than 11 daily orbits.

LEO is considered near enough to Earth for convenient launch, communication and observation. It also is the area where the International Space Station currently orbits and where many proposed future platforms will be located, according to NASA.

An orbit for a U.S.-launched Amateur Radio satellite is coordinated through a multi-agency and international process. A non-governmental organization, the International Amateur Radio Union ([IARU](#)), handles frequency coordination, while U.S. government agencies such as the Federal Communications Commission ([FCC](#)) and National Telecommunications and Information Administration ([NTIA](#)) oversee licensing and national-level coordination. AMSAT volunteers can assist educators in the process of obtaining an orbital assignment.

The AMSAT [CubeSat Simulator](#), described in detail later in this Guide, is a relatively low cost teaching tool, available on loan to AMSAT [Ambassadors](#) for use in schools as a teaching tool. It is nearly identical to a completely functional space satellite. It is not flight-ready for launch into space but functionally the same.

Outer space hardware even for LEO satellites must be flight-ready and meet stringent requirements. These components cost more. Flight-ready hardware must meet rigorous specifications and undergo testing before it can be launched into orbit.

Launching a satellite into a LEO usually is ***cost-effective***. Less costly observation systems (e.g. onboard cameras) can produce higher resolution images because the satellite is closer to Earth. In educational and/or scientific research context LEO

satellites are easier and more economical to launch, requiring less energy and less powerful radio transmitters for successful communication.

Radio signals play a critical role in the life or death of nearly all satellites in orbits near and far. Data about satellite functions is sent from the satellite to specially outfitted ground stations. The antenna and radio equipment receives and transmits telemetry data. A ground station uses computer software to decode the telemetry data.

The telemetry data provides measurements of satellite electronic components and various physical and electronic subsystems. The ground station radio operator shares telemetry data with project engineers who can evaluate the health of a satellite and take corrective actions using [telecommands](#) to the satellite from its [on-board data handling](#) system.

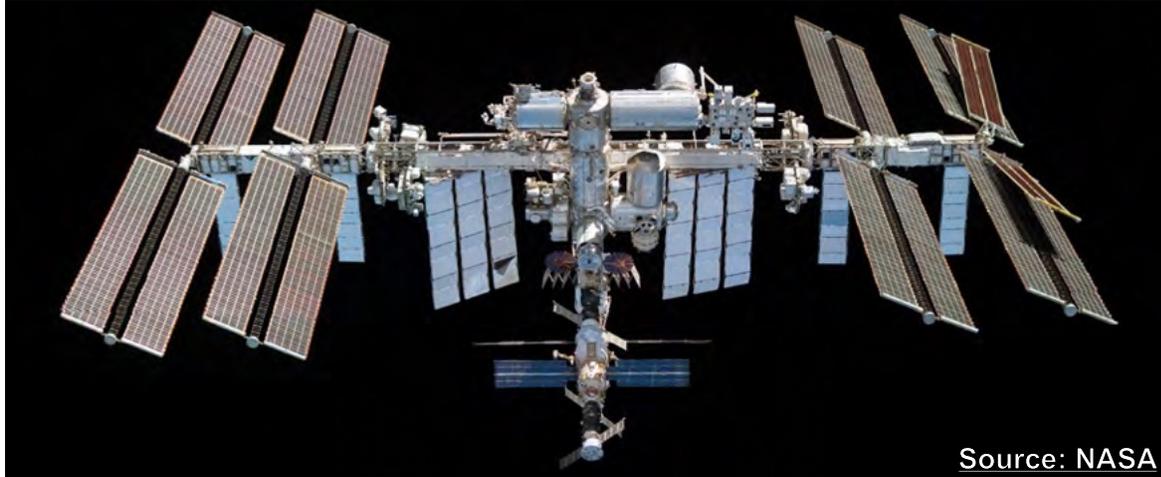
One drawback to a low earth orbit is the length of time a ground station can communicate with the satellite. Faster velocity and lower altitude mean that the satellite is within range of the ground station radio for a short time period. The good news is that another suitable pass of the satellite may occur repeatedly on the same day.

Mathematics is key to understanding [orbital mechanics](#). This subject includes understanding and calculating the motion of objects in space, such as satellites and spacecraft, under the influence of gravity and other forces like atmospheric drag. It is a core discipline in aerospace engineering that applies Newton's laws of motion and universal gravitation to design space missions and control spacecraft trajectories.

The ideal radio telemetry reception and transmission angle is a relatively narrow arc before and after the satellite is directly overhead. More often the orbit is not overhead so an even shorter 5 to 10 minute “window” for communication is available to a radio system at a particular location. Two-way radio contact is possible only when the satellite is sufficiently above the horizon for [line of sight](#) transmission.

For additional information about scientific benefits see the [Annual LEO Science Digest](#) which highlights the broad applications and benefits of low Earth orbit (LEO) observations. Also see NASA's [LEO FAQ](#) web page.

The Space Station – Classroom Connection



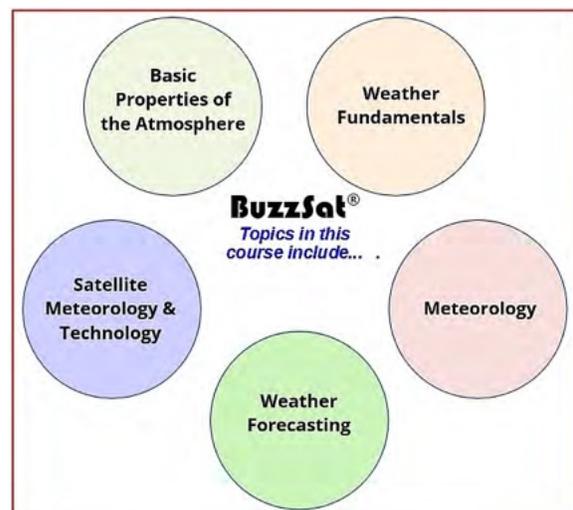
The International Space Station ([ISS](#)) is arguably the largest satellite orbiting Earth at the moment. [Science](#) is at the core of ISS activity. According to NASA, “Orbiting more than 200 miles (320 km) above Earth, the International Space Station is a powerhouse of cutting-edge science that is unlocking discoveries not possible on Earth. We’re testing technologies that are critical to our return to the Moon and contributing to medical and social breakthroughs that improve life on our home planet.”

Students can take part in the ISS activities via Amateur Radio on the International Space Station ([ARISS](#)) program. ARISS provides selected K-12 schools from around the globe a space science oriented curriculum with the capstone being live discussion with an astronaut aboard the ISS.”

AMSAT Teaching Tools for Educators

[BuzzSat.com](#) is an AMSAT learning center focused on how satellites improve our daily lives. An extensive online course is available free of charge. The site offers an [Introduction to Satellite Meteorology](#) self-paced instructional course. The course explores the many ways that satellites help meteorologists predict the immediate weather and long-term climate behavior to make our lives more enjoyable and safer.

This “self-standing” course is designed for students in grades 8-12. No background in science or math is required. Any student, parent, youth leader or teacher may



independently utilize the course. Each chapter has 40-50 short web pages and should take a user about 30 minutes per chapter. Most pages are illustrated and there is a quiz system built into the course. A glossary is available to the users throughout the course.

Current and future courses will demonstrate the concept: "Satellites in Space Help Us Live a Better Life on Earth."

AMSAT CubeSat Simulator

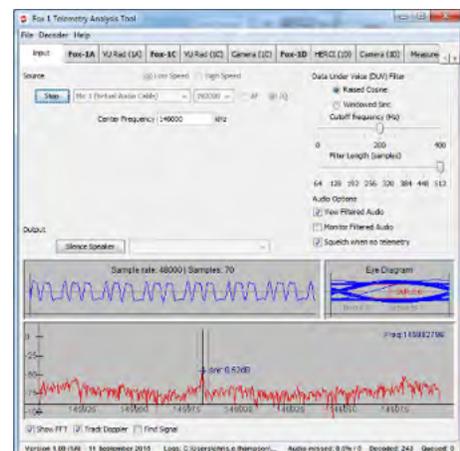


The [CubeSatSim](#)[™] project is headed by AMSAT VP, Educational Relations [Alan B. Johnston](#), PhD, KU2Y of the Villanova University's Electrical and Computer Engineering Department. The CubeSatSim is a low-cost satellite emulator that runs on solar panels and batteries. It transmits low power UHF radio telemetry on Amateur Radio frequencies. Available as a parts kit or a no-solder kit, the CubeSatSim has a 3D printed frame and can be extended by additional sensors and modules.

Radio transmissions like telemetry require software to decode and understand what is being transmitted. For example, Amateur Radio satellite telemetry can be decoded using a software program called [FoxTelem](#).

"FoxTelem" refers to the software and data associated with the [AMSAT Fox series of CubeSat](#) satellites.

This free and open source software is used to demodulate, display, analyze and store the [telemetry data](#). Telemetry from AMSAT CubeSat spacecraft is transmitted via Amateur Radio signals to ground-based radio



receivers worldwide. FoxTelem software runs on Windows, Linux, or inexpensive [Raspberry Pi](#) single board computers.

A good first step in learning about FoxTelem is the AMSAT [Fox-in-a-Box](#) (FIAB) free Raspberry Pi software. Conveniently self-contained, the disk image includes FoxTelem and sample audio files of actual telemetry signals so no radio is necessary. Read setup instructions in this [PDF file](#). The AMSAT YouTube channel includes a playlist of three [videos](#) about FIAB.

The FIAB Raspberry Pi Image also contains the following open-source software:

- [OpenWebRX WebSDR](#)
- [SSTV Decode](#) using [QSSTV](#)
- [SDR++](#)
- [Direwolf](#)
- [CubicSDR](#)
- [RTL-TCP](#)
- [FunCube Telem](#)
- [Gpredict](#)
- [PacSat](#)
- [G0KLA Tracker](#)
- [KlaTrack](#)
- [AMSAT Pacsat Ground Station](#)

Worldwide, millions of people monitor earthbound and space-based radio transmissions. This includes [scientists](#), radio monitoring [specialists](#) and [hobbyists](#) who use inexpensive [SDR](#) (Software Defined Radio) USB devices and many different open source [software](#) programs. Good quality SDR USB dongle radio receivers cost as little as \$30 ([RTL-SDR](#)). More complex [SDR systems](#) are also used.

In some areas that afford a wide-open view of the sky especially open areas at higher-than-average terrain, a simple hand-held radio or SDR receiver can be used to listen to satellite transmissions. If there is a licensed Radio Amateur operator supervising your use of a transceiver you may be able to speak to an astronaut aboard the ISS or make contact with another Radio Amateur operator via the satellite.

For optimum reception of radio signals an outdoor [antenna](#) is usually required. An outside antenna is needed for satellite radio telemetry reception. There are many types of [antennas](#) designed for satellite telemetry reception, both Do it Yourself (DIY) and commercially manufactured.

NOTE TO EDUCATORS: Loaner kits are available. See this [video](#) for more information and/or read this [Loaner Guide](#) for more details.

Why Study Telemetry?

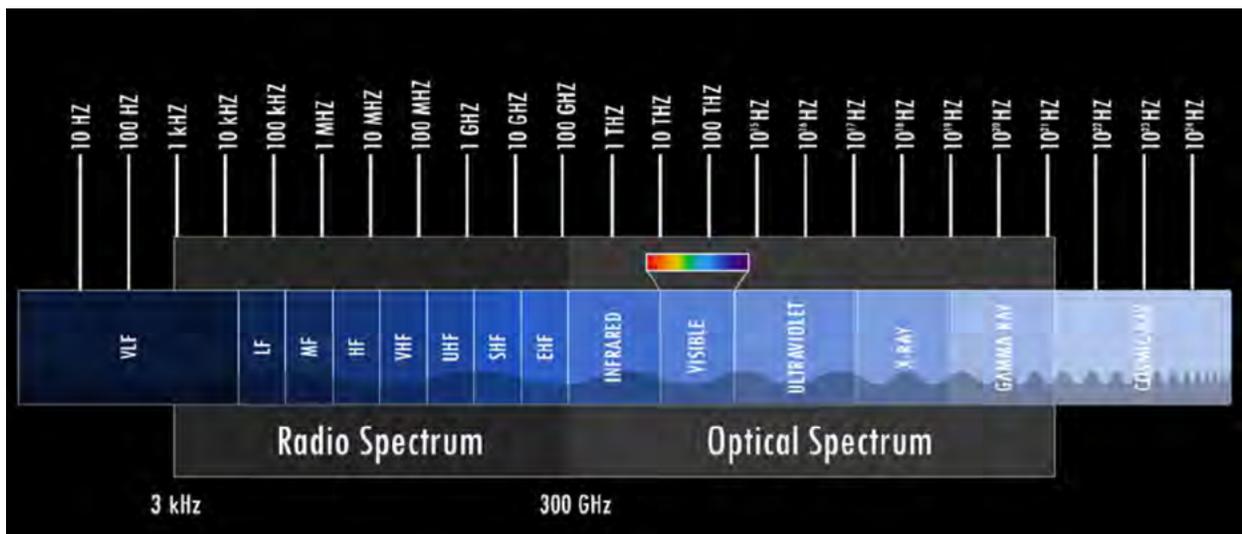
Students should know that many well-paying jobs (both professional and technical) in computers and technology [employment](#) require a knowledge of [telemetry systems](#) to perform their duties. For [example](#) jobs in medical, research, telecommunications and aerospace often require some knowledge of telemetry.

Telemetry is all around us, keeping us safe and healthy. For example:

- National Security -- Threat identification and monitoring.
- Hospitals – Patient monitors to inform and alert nurses and doctors.
- [Meteorology Satellites](#) – Weather forecast and warnings of dangerous situations ([PDS](#)).
- Aviation – Automatic Dependent Surveillance–Broadcasts ([ADSB](#))
- Petroleum Pipelines-- Routine monitoring, flow control and [leak detection](#).
- Natural Gas Pipelines – Integrity Management of Gas Transmission Pipelines ([PDF](#)).
- Electric Utilities – Fault detection in electricity generation and transmission. ([PDF](#)).
- Internet of Things devices ([IoT](#)) -- Vast array of convenience, maintenance and safety purposes

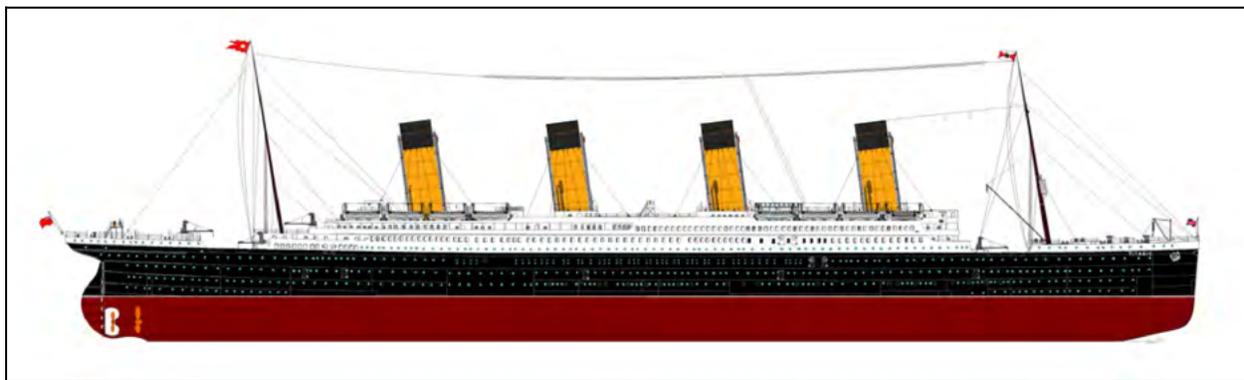
Exploring Radio Is Fun

Science teaches us about the [Electromagnetic Spectrum](#) (EMS) which includes [radio frequencies](#). Radio waves must be [modulated](#) to carry information: voice, images or data, etc. Commonly used radio frequencies, such as those for AM and FM radio broadcasts, are a small portion of the electromagnetic spectrum. [Image source](#). Physics students can be fascinated by an endless variety of topics about the EMS.



Radio frequencies are measured in cycles per second (cps), mostly in multiples of 1,000 cps. Multiples of cps are labeled kilo (a thousand), Mega (a million) and Giga (a billion). Radio frequencies are signified by [Hertz](#) (Hz) with a multiplier designation such as kHz, MHz or GHz. These measurements are [derived](#) within the International System of Units ([SI](#)).

The simplest way to communicate information by radio is to turn the radio wave on and off to create the dots and dashes of Morse Code. Students interested in history might be interested to know that radio telegraphy was the earliest means of transmitting information by radio. It was first used by ocean going ships not long after [Guglielmo Marconi](#) demonstrated radio in 1901.



When the luxury ocean liner [Titanic](#) was sinking on April 15, 1915 radio telegraphy was used to summon assistance from other ships within range of the transmission. This early tragedy galvanized the public's attention and [accelerated growth](#) in the field of radio. Drawing is courtesy of [Lux's Type Collection, Ocean liners](#).

Today's sophisticated radio transmitter-receiver, or "transceiver" (hand-held, mobile or base station) can transmit information using many types of [modulation](#). Nearly all Amateur Radio transceivers have a voice mode but more expensive radios can transmit a variety of digital modes as well, primarily in the Very High Frequency (VHF) band, 144 to 148 MHz +/- and Ultra High Frequency (UHF) band, 420 to 450 MHz +/-.

Amateur Radio

In the U.S., the Federal Communications Commission (FCC) grants an [Amateur Radio Service](#) license dependent upon passing exams administered by [Volunteer Examiners](#). Exams are administered in person and using through the use of specially designed video teleconferences for individuals. Amateur Radio license levels include Technician, General and Extra, each dependent upon passing increasingly difficult [examinations](#). A basic level of knowledge is needed to pass a Technician license test

but increasingly greater knowledge is required for successful completion of General and Extra class licenses.

Radio Amateurs are uniquely fortunate to be able to use a broad range of radio frequencies and methods of communicating voice and data depending upon the level of license achieved. All levels of license permit the use of Amateur Satellite communications.

Dependent upon license class, Radio Amateurs may transmit on many narrow segments of radio frequencies from below the AM radio broadcast band into the high microwave region of radio frequencies. In addition to radio frequency access privileges, license holders may use an astonishingly wide variety of transmission modes, both analog and digital. There are many other [radio modulation modes](#) authorized by the FCC.

So far, Amateur Satellites use one or more of a handful of different transmission modes. However, depending upon the license class, Radio Amateurs use many different types of radio transmissions. You can even send email via Amateur Radio or transmit video to fellow Radio Amateurs.

The Amateur Radio Relay League ([ARRL](#)) is the National Association for Amateur Radio.® Many different services are available to anyone via the internet. An ARRL [directory](#) lists affiliated local clubs and groups. Also, School and [College](#) Clubs may be available in your locality.

Amateur Radio includes fun, family, friends, public service and STEM (Science, Technology Engineering, and Math) education. In the realm of emergency communications “When All Else Fails, Amateur Radio” is the operative description applicable to many organized groups of Radio Amateurs.

Some Radio Amateurs operate from designated parks (local, county, state, national) in the “Parks On The Air” (POTA) program. POTA activities have become increasingly popular during the past few years. A similar program, “Summits On The Air” (SOTA) is also popular.

Radio Amateurs travel to exotic destinations throughout the world on what is referred to as [DX-Peditions](#) so other Radio Amateurs can record a contact from regions of the world where there are few if any indigenous Radio Amateurs.

These and other activities for Radio Amateurs are collectively called “[Radio Sport](#).”

Licensing, operation and opportunities for public service are just some of the activities for young Radio Amateurs. There are many activities organized exclusively for younger Radio Amateurs. Also, hundreds of scholarships for youthful Radio Amateurs are awarded throughout the U.S. These programs rely on the traditional generosity of Radio Amateurs which is a hallmark of The Amateur Radio Service.

Future Employment Opportunities

Career counselors should understand that obtaining an Amateur Radio Service license grant from the [FCC](#) demonstrates personal achievement as well as the specialized knowledge required to pass the license examination. In turn, this widely recognized certification provides a pathway to many exciting and good paying career opportunities. in various [electronics engineering](#) and a wide variety of [technology-oriented](#) and even academic [research](#) or teaching fields.

“The future of the economy is in [STEM](#),” says James Brown, the executive director of the STEM Education Coalition in Washington, D.C. “That’s where the jobs of tomorrow will be.”

Employment [opportunities](#) in Space Communication Technology are diverse, spanning roles like Satellite Engineers, RF (Radio Frequency) Engineers, Systems Engineers, and Ground Station Operators who design, build, and manage satellite systems. Other roles include Satellite Operators, Network Analysts, Cybersecurity Specialists, Software Developers, and Project Managers, with opportunities in [government](#) agencies, [military](#), and private [companies](#) like those in the aerospace and telecommunications sectors.

NOTE TO EDUCATORS: ARRL provides several programs specifically for schools and colleges. Visit their [Outreach](#) web page more details.



Home

Edit

New page

[Jump to bottom](#)

Alan Johnston edited this page 1 minute ago · [159 revisions](#)

Welcome to the AMSAT® CubeSatSim Project Wiki, the CubeSat Simulator

The CubeSatSim(TM) is a low cost satellite emulator that runs on solar panels and batteries, transmits UHF radio telemetry, has a 3D printed frame, and can be extended by additional sensors and modules. This project is sponsored by the not-for-profit [Radio Amateur Satellite Corporation, AMSAT®](#).

This page is for the new v2.0 hardware (blue boards, v2.0 and later).



If you have the v1.2 hardware (white PCBs) or earlier, then you need these wiki pages: <https://CubeSatSim.org/wiki-v1>

If you have the beta v1.3 hardware (blue PCBs), the v2.0 instructions are almost the same, but you can use these wiki pages: <https://CubeSatSim.org/wiki-beta>

Here are some **Quick Start Guides** for the operation of the CubeSatSim <https://CubeSatSim.org/qsg>

The Bill of Materials (BOM) is here:

<https://cubesatsim.org/bom>

NOTE: If you use the Octopart links in the BOM, the Digikey parts will have the step and identifier information printed on them.

The hardware information including Gerbers is available at:

<https://github.com/alanbjohnston/CubeSatSim/tree/master/hardware/v2.0>

Here you will find documentation about this project and detailed install instructions.

Features

The CubeSatSim has the following features:

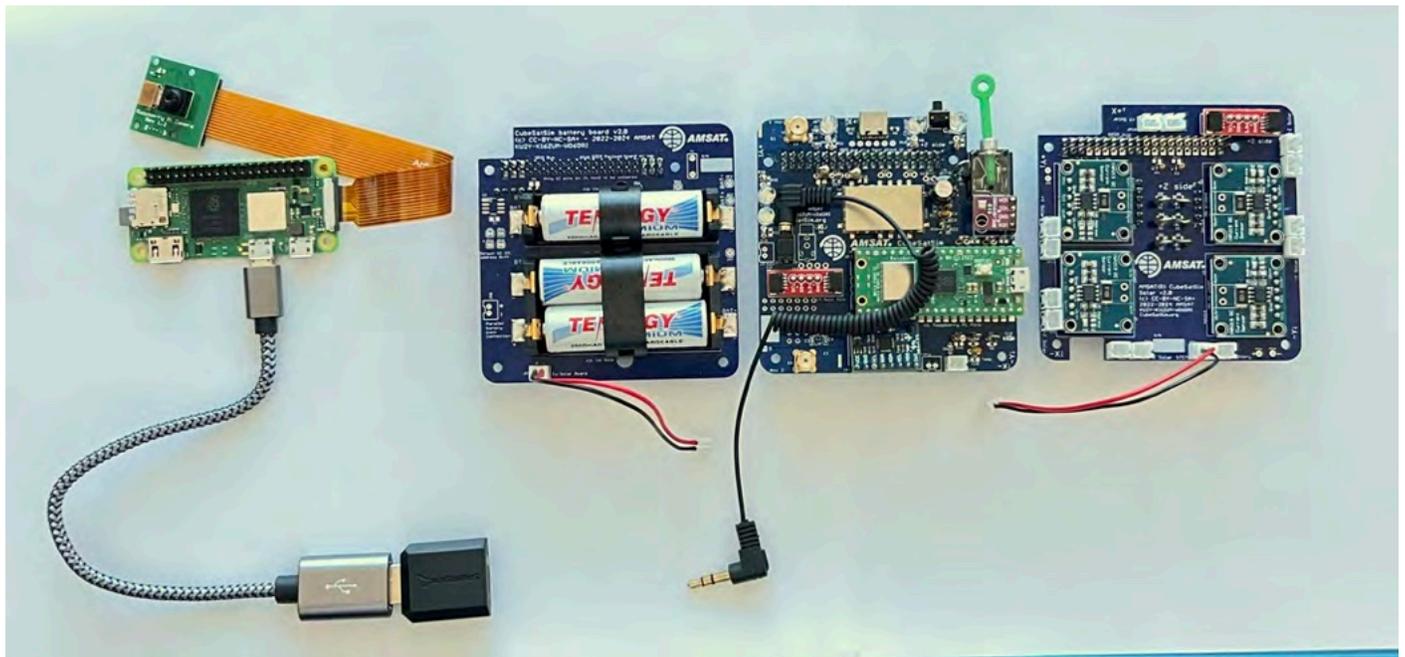
- Working solar panels and rechargeable batteries
- Multi-channel voltage, current, and temperature telemetry transmitted in the Amateur Radio UHF band
- Telemetry decoding using [FoxTelem software](#) or APRS software
- Payload microcontroller Raspberry Pi Pico and sensors
- Tape measure monopole, dipole, or SMA antenna
- Integrated Low Pass Filter
- 3D printed frame and solar panels

Here are the changes between the v1.2 and the v2.0 hardware and software.

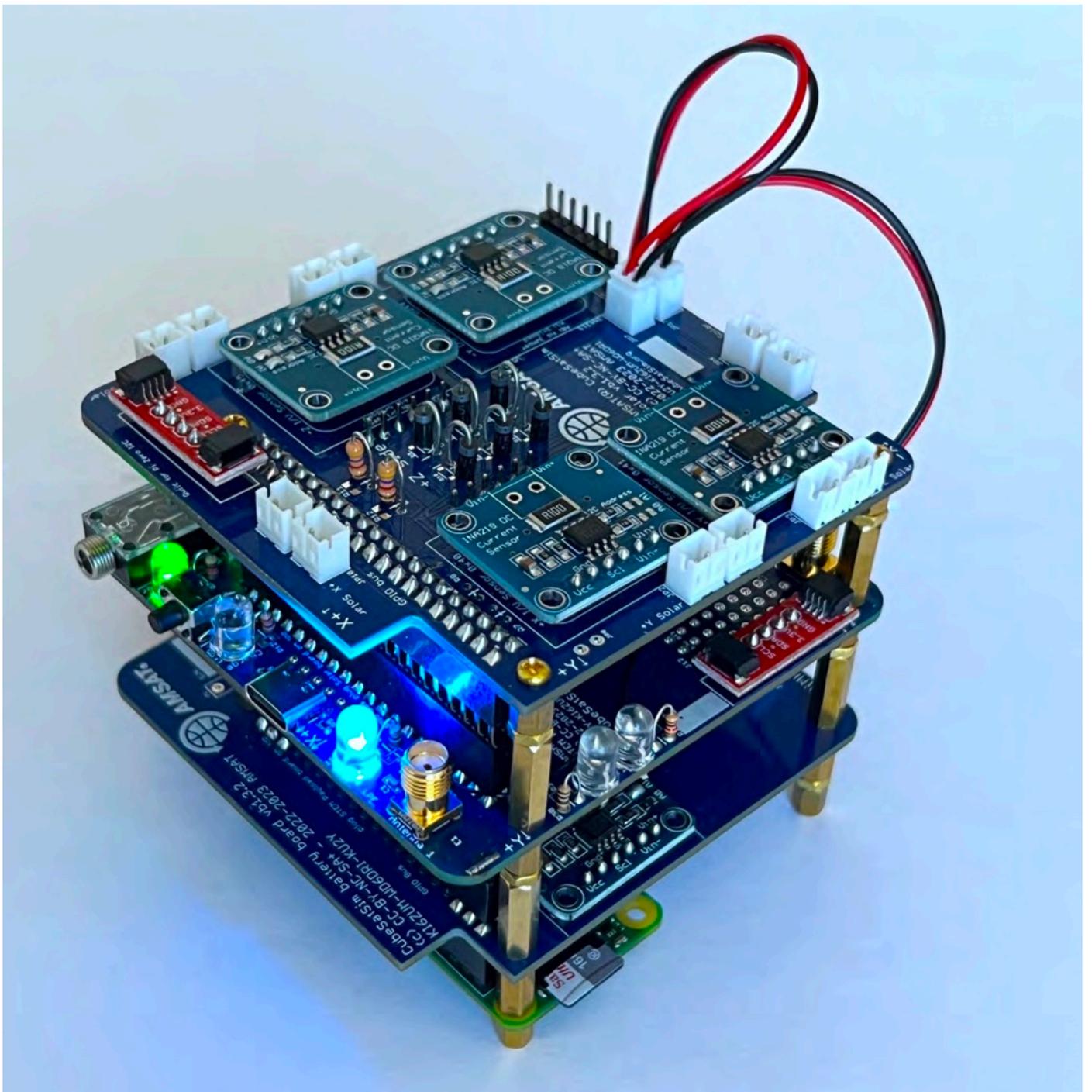
- New FM transceiver module for better frequency stability and simple command and control receiver to change telemetry mode
- More modern and cheaper Raspberry Pi Pico micro controller
- Easily connect additional sensors for the Pico using the Qwiic connector system <https://www.sparkfun.com/qwiic>
- SSTV camera images now display callsign and battery status overlay

- Can be modified to fly as a balloon payload with 500mW FM output for SSTV, APRS, or CW transmissions with software support on Pico for a serial GPS module
- Lower parts cost and easier to source. All parts can now be sourced from electronics distributors and Amazon including easy to find solar panels. New BOM uses Octopart electronic part inventory site with one click distributor ordering
- Redesigned for blue INA219 voltage and current sensors instead of more expensive purple ones
- Battery board now has integrated voltage and current sensor and stronger holder with better performance
- Simpler electrical power system with no boost converter or charge control circuit
- Kits can be built with through hole parts except for a few SMT parts. Fully assembled boards will be available in the future using SMT parts
- Easily connect additional sensors for the Raspberry Pi Zero the Qwiic connector system

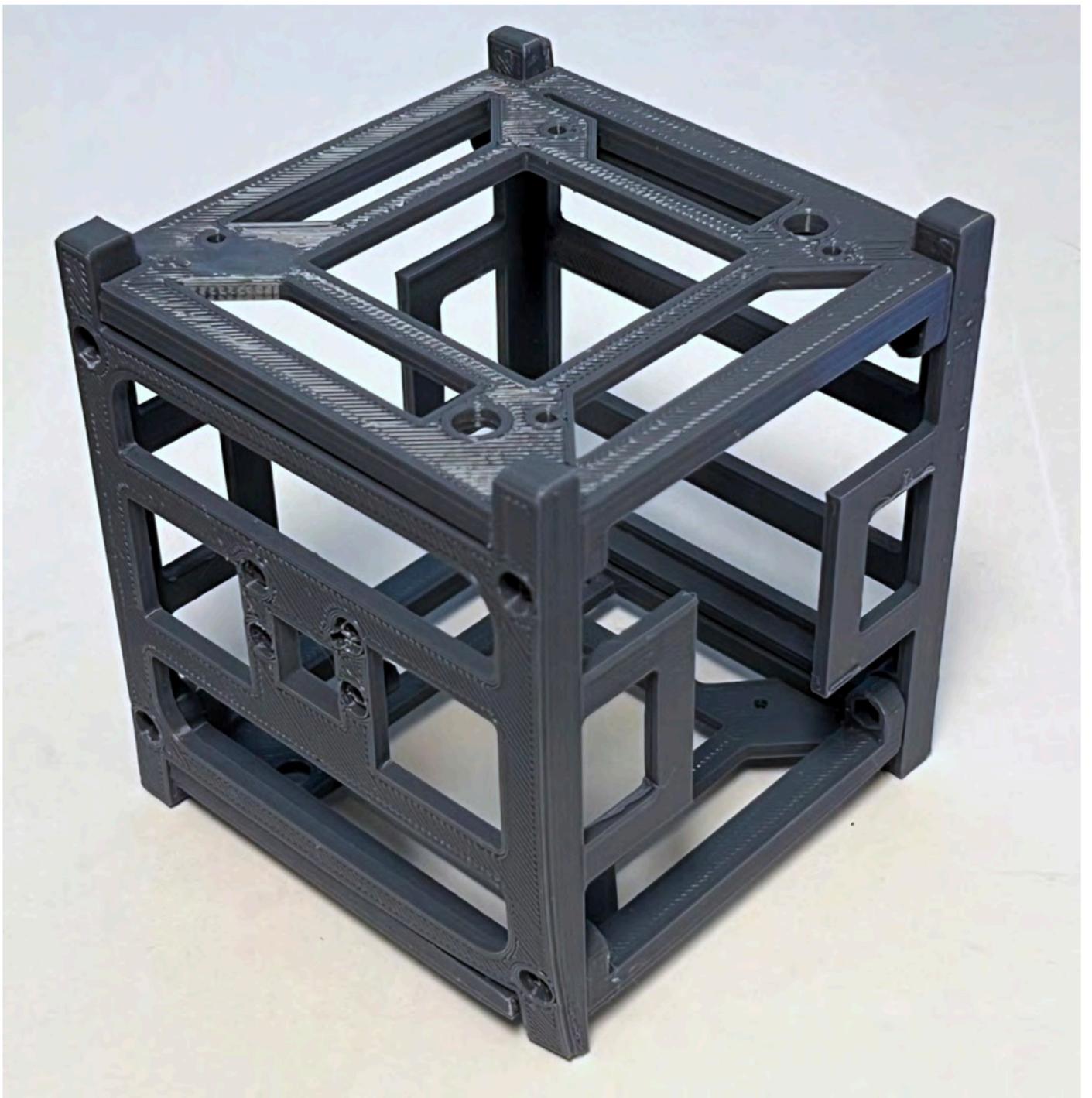
Here are the four boards that make up the complete board stack. Left to right: Raspberry Pi Zero WH with Pi Camera, Battery Board, Main Board with FM module, and Solar Board.



Here is the built board stack:



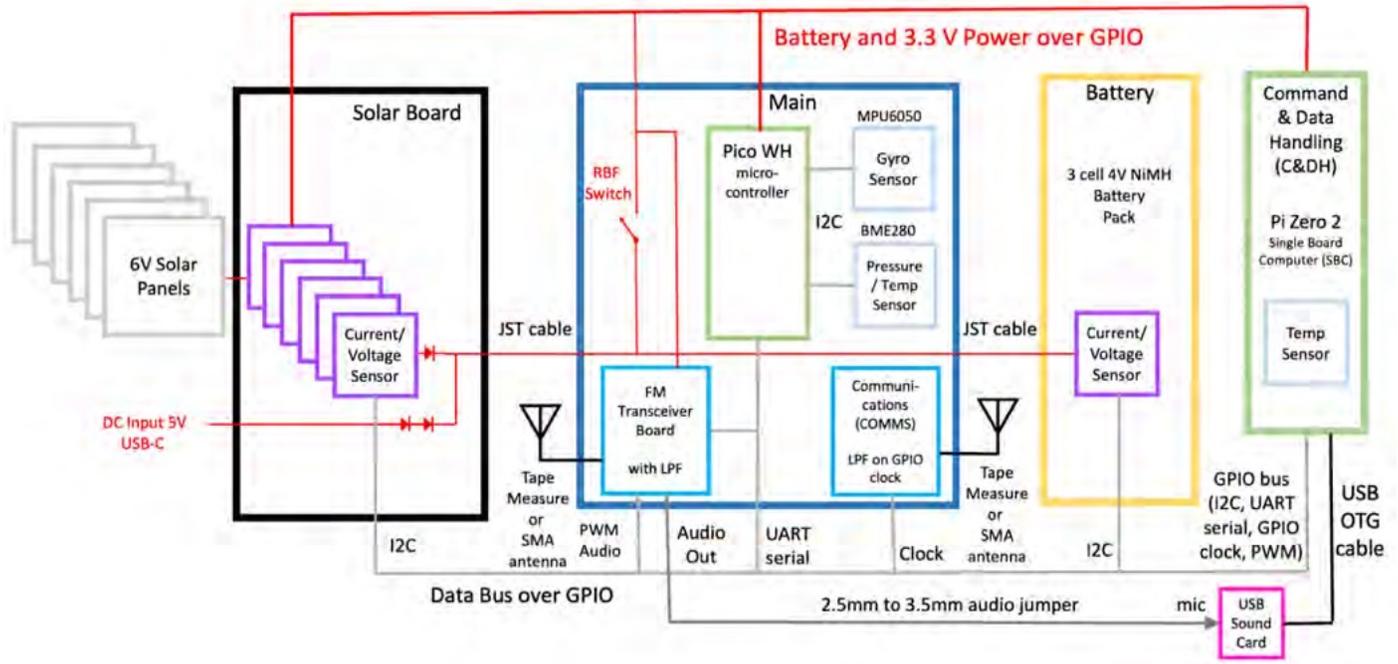
There is a 3D printed frame:



Block Diagram

Here is a block diagram of the design:

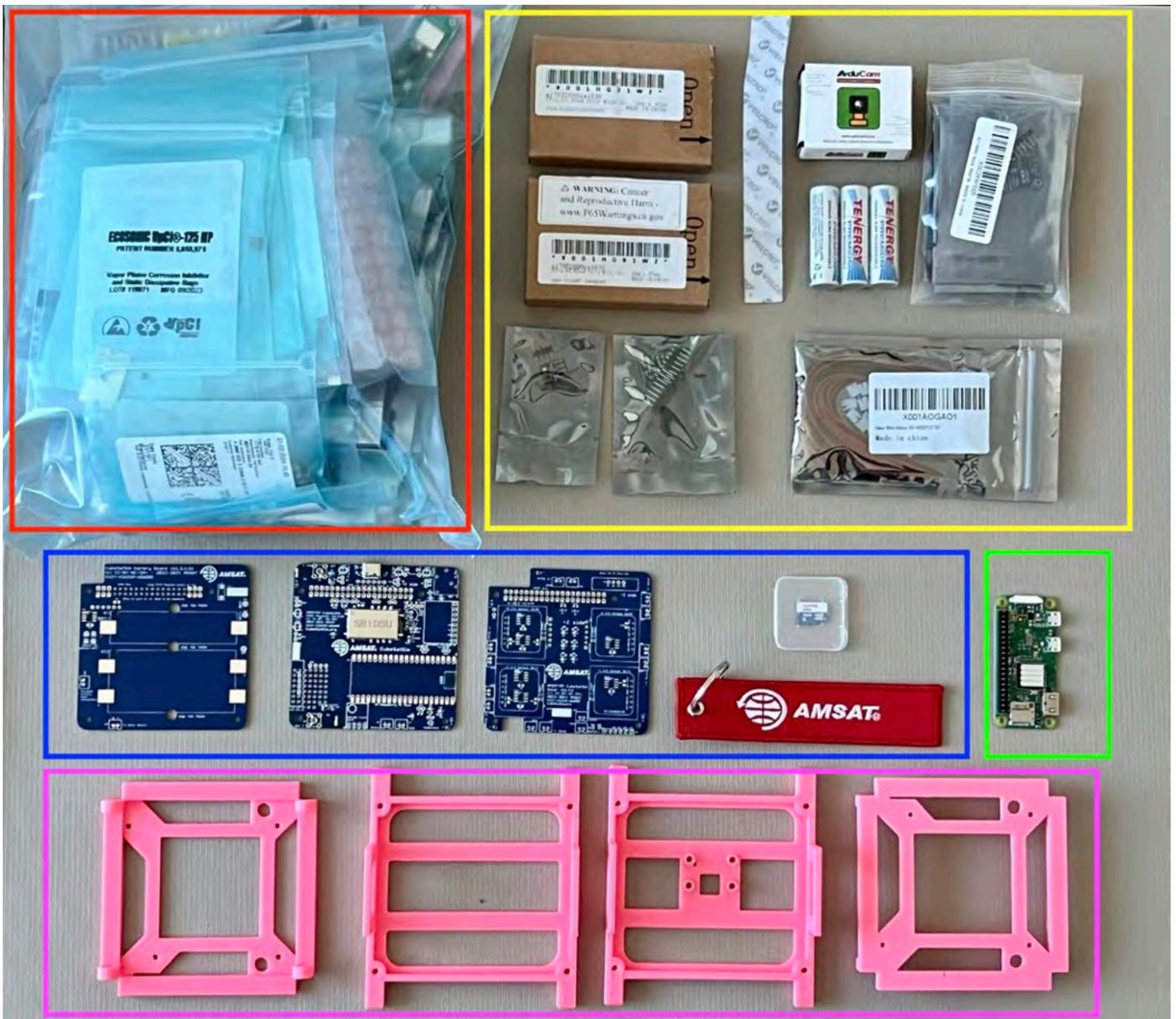
CubeSatSim Block Diagram v2.0



Bill of Materials

Parts List Bill of Materials (BOM) to build the CubeSatSim is available here. Make a copy of this spreadsheet: <https://CubeSatSim.org/bom>

Here is a photo of a kit of parts. Boxed in red are the parts from Digikey, yellow parts from Amazon, blue parts from AMSAT, green parts from various places (use rpilocator.com to find in stock Pi Zero WH), and pink is the 3D parts you print yourself:



Build Steps

Here are the steps to build a CubeSatSim:

- [1. Build the Main Board Part 1](#)
- [2. Install the Software](#)
- [3. Build a Ground Station](#)
- [4. Continue Building the Main board Part 2](#)
- [5. Build the Battery board](#)

[6. Build the Solar board](#)

[7. Assemble the Solar Panels and Camera](#)

[8. Put the Board Stack together and mount in the Frame](#)

[9. Final Testing](#)

Here is this [page in PDF](#)

Here is a single 35MB PDF with all the [Wiki Instructions for the CubeSatSim](#)

Glossary

If you are new to Amateur Radio Satellites, here is a handy [Glossary](#) with common terms and acronyms.

+ Add a custom footer

▼ **Pages** 162

Find a page...

▼ **Home**

Welcome to the AMSAT® CubeSatSim Project Wiki, the CubeSat Simulator

Features

Block Diagram

Bill of Materials

Build Steps

Glossary

▶ **1. Main Board 1**

▶ **2. Software Install**

▶ **3. Ground Station**

▶ **4. Main Board 2**

▶ [5. Battery Board](#)

▶ [6. Solar Board](#)

▶ [7. Solar Panels and Camera](#)

▶ [8. Board Stack](#)

▶ [9. Final Testing](#)

▶ [Adding New Sensors](#)

▶ [Beta Upgrade to v2.0](#)

▶ [Command and Control](#)

▶ [Creating the CubeSatSim Raspberry Pi Image](#)

▶ [Cross Band Repeater](#)

Show 147 more pages...

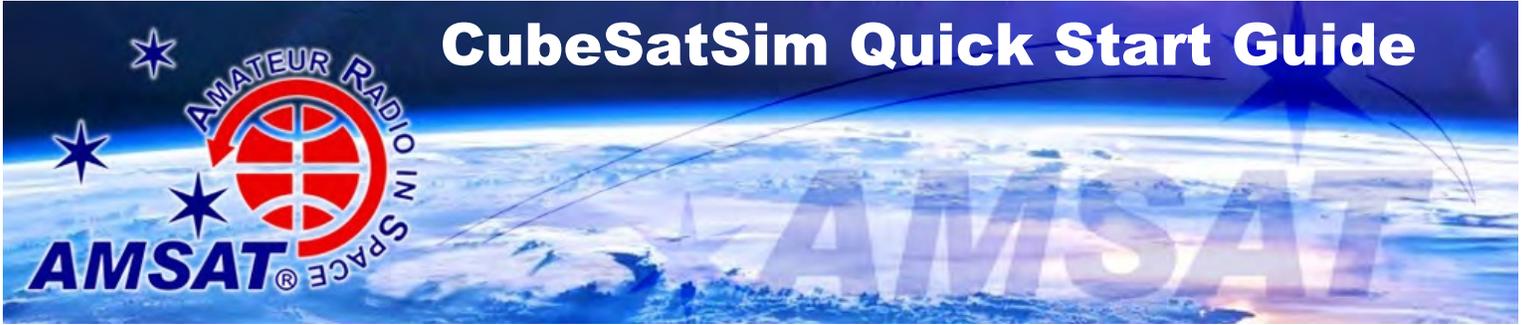
+ Add a custom sidebar

Clone this wiki locally

<https://github.com/alanbjohnston/CubeSatSim.wiki.git>



CubeSatSim Quick Start Guide



How to Turn on the CubeSatSim:

If the Remove Before Flight pin is plugged in (see photo), unplug pin and the CubeSatSim should turn on in about 30 seconds. The Green Power LED will first turn on, then the Blue Transmit LED will blink when it is transmitting.

If removing the Remove Before Flight pin doesn't turn it on, try inserting and removing the Remove Before Flight pin. If it still doesn't turn on, the battery may be drained – see below on how to charge the CubeSatSim.

How to Turn Off the CubeSatSim:

Press and hold the pushbutton. The Green Power LED will blink a number of times fast, then it will blink slowly. Release the pushbutton as it is blinking slowly and the CubeSatSim will begin to shut down. After about 20 seconds, insert the Remove Before Flight pin to completely shut down the CubeSatSim.



How to Charge the CubeSatSim:

The CubeSatSim can be charged while turned on or off, with the Remove Before Flight pin unplugged or plugged. For fastest charging, charge with CubeSatSim off and the Remove Before Flight pin plugged in.

Plug the USB-C cable into the USB charging connector. The Red Charging LED will illuminate. Charging for 8 hours is needed for a full charge. The batteries will self-discharge so charging should be done the day before or night before use.

If the Remove Before Flight pin is not inserted during storage, the batteries will quickly drain, so always insert the Remove Before Flight to shut down the CubeSatSim.

This file can be downloaded here: <https://cubesatsim.org/qsg>
Videos are available at <https://cubesatsim.org/qsg-videos>



How to Receive Radio Signals from the CubeSatSim:

The CubeSatSim transmits on 434.9 MHz (or 434900 kHz). The Blue Transmit LED will be illuminated when it is transmitting. The frequency can be offset slightly from this frequency, so you may need to tune up or down a few kHz.

When the CubeSatSim starts up or changes modes, it will transmit a CW ID (Morse Code) that is HI HI DE CALLSIGN where CALLSIGN is either the default AMSAT or the callsign you configured on the Raspberry Pi Zero 2.

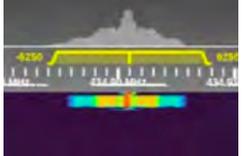
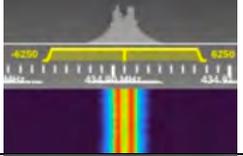
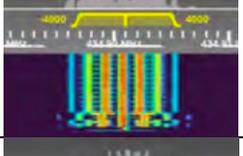
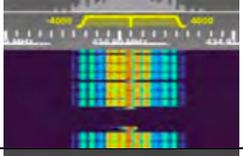
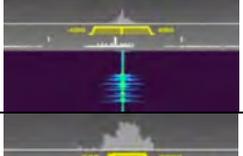
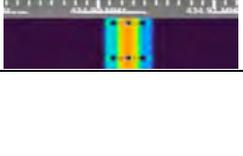
You can listen to it on a radio or computer with an SDR (Software Defined Radio) RTL-SDR USB dongle plugged in.

How to Change the CubeSatSim Telemetry Mode:

Each of the seven mode generates a different type of radio transmission. You can hear the difference between the different modes, and see a difference on your computer waterfall if you are using an SDR.

To change modes, press and hold the Pushbutton while the Green Power LED blinks. If you release the Pushbutton as it is blinking, the CubeSatSim will change to the mode associated with that number of blinks (see Mode Table).

The Mode Table shows each mode and how many blinks are used to set that mode. Command tells you the **CubeSatSim/config** command that can be typed into the Raspberry Pi Zero 2 to change to that mode. Decoding tells you what software you need. Note that some modes also require SDR software and audio loopback software as well. Audio is a link to an audio file so you can hear what the mode sounds like. Waterfall shows how the signal looks in an SDR.

Blinks	Mode	Description	Decoding	Command	Audio	Waterfall(WebSDR)
1	APRS	Automatic Packet Reporting System. This digital mode sends a packet of data with AFSK or Audio Frequency Shift Keying modulation.	Windows: SoundModem or Direwolf Raspberry Pi/Linux: OpenWebRX or Direwolf	a	CubeSatSim.org/a	
2	FSK	Frequency Shift Keying. This mode transmits a continuous signal that makes a rumbling sound that emulates the AMSAT Fox CubeSats such as Fox-1C or AO-95. Also known as DUV or Data Under Voice.	Windows/Raspberry Pi/Linux: FoxTelem	f	CubeSatSim.org/f	
3	BPSK	Binary Phase Shift Keying. This mode transmits a continuous signal that sounds like noise that emulates the AMSAT Fox-1E CubeSat. You demodulate using USB.	Windows/Raspberry Pi/Linux: FoxTelem	b	CubeSatSim.org/b	
4	SSTV	Slow Scan TeleVision. This mode transmits stored images in Scottie 2 format which sounds like a series of tones.	Windows: MMSSTV Raspberry Pi/Linux: QSSTV	s	CubeSatSim.org/s	
5	CW	Continuous Wave or Morse Code. This mode transmits a FM modulated tone at 20 words per minute Morse Code telemetry.	Windows/Raspberry Pi/Linux: fdigi with spreadsheet http://cubesatsim.org/telem	m	CubeSatSim.org/m	
6	Cross Band Repeater	Simulates a U/V (UHF uplink, VHF downlink) FM satellite repeater. Default uplink is 435 MHz and downlink is 144.9 MHz.	OpenWebRX	e	CubeSatSim.org/e	
7	Fun Cube	FUNcube simulates the BPSK telemetry of the FUNcube series of CubeSats such as AO-73.	Raspberry Pi: FUNcube Telem	j	CubeSatSim.org/j	

AMSAT AMATEUR RADIO IN SPACE Ground Station Quick Start Guide

This guide provides a short description of how to use the CubeSatSim Ground Station. The Ground Station is an SDR (Software Defined Radio) receiver that has the capability of receiving and decoding signals from the CubeSatSim. For details of assembly of the Ground Station, please see: https://www.youtube.com/watch?v=l1o9LD_2hIU For details of assembly of the CubeSatSim please see: <https://github.com/alanbjohnston/CubeSatSim/wiki>

Before you begin, please make sure that you have the following ready:

- [Raspberry Pi version 3B or later \(Pi 4 or 5 is fine\)](#)
- [RTL-SDR Receiver](#) with an antenna
- [AMSAT ARISS Pi image](#) loaded on a Micro SD card and inserted into the Raspberry Pi
- [Power Adapter for the Raspberry Pi](#)
- Keyboard and mouse connected (you can use a [mini keyboard/mouse](#) device)
- HDMI Monitor



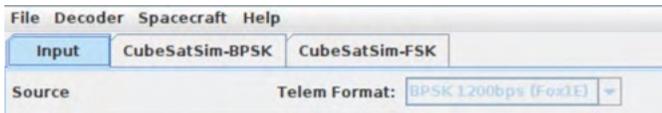
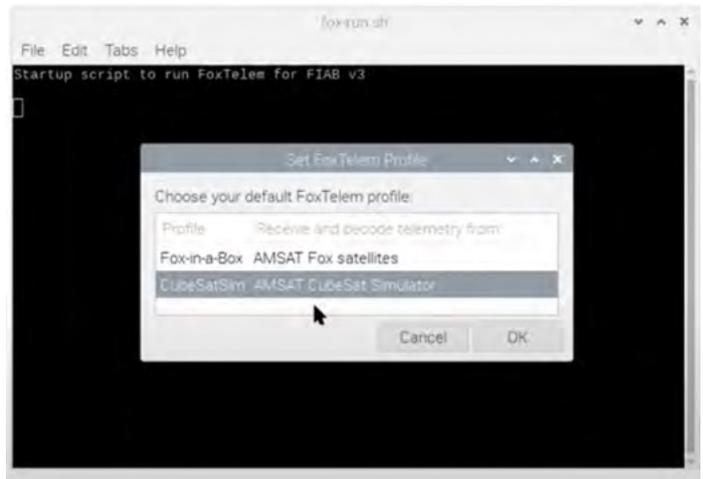
Steps:

1. Turn on your CubeSatSim. see <https://cubesatsim.org/download/cubesatsim-qsg.pdf> (make sure you are in BPSK mode)
2. Make sure your RTL-SDR, keyboard/mouse, and monitor are inserted in Raspberry Pi and then apply the power. It will take about 10 seconds for the operating system to load and then FoxTelem will automatically load (wait for it to load completely).
3. When you run it the first time, FoxTelem will ask you which satellite profile you want to use - choose "CubeSatSim AMSAT CubeSat Simulator"

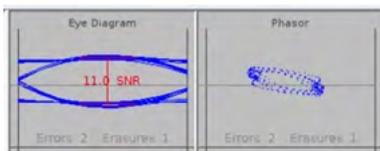
To Decode BPSK:

The CubeSatSim will initially start in BPSK Mode, or you can change to mode 3 using the pushbutton (three blinks of the Green LED).

FoxTelem will start listening in BPSK Mode (Mode 3) and automatically tuned to around 434.9 MHz. The "Telem Format" will be "BPSK 1200bps (Fox1E)"



If you have a handheld VHF FM radio, you can listen on that frequency for the sound. FoxTelem will then open to the "Input" tab that shows the waveforms of the signal.



If the "Eye Diagram" looks like an eye, you will start to see an increasing count of frames of data being decoded in the lower left corner of the window.

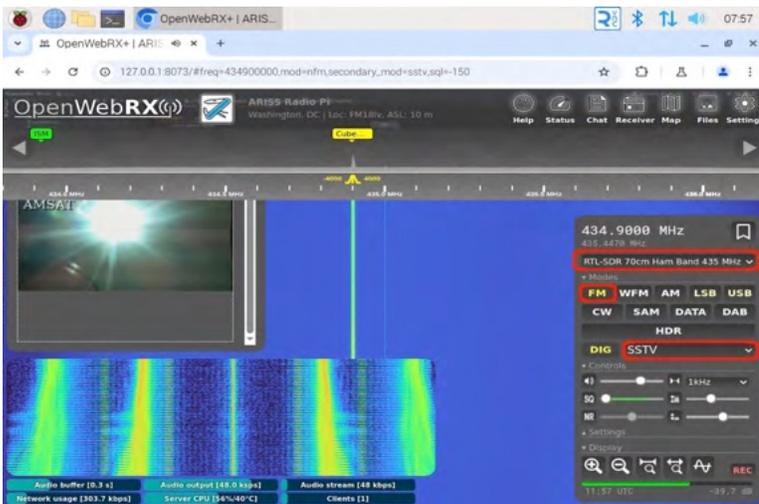
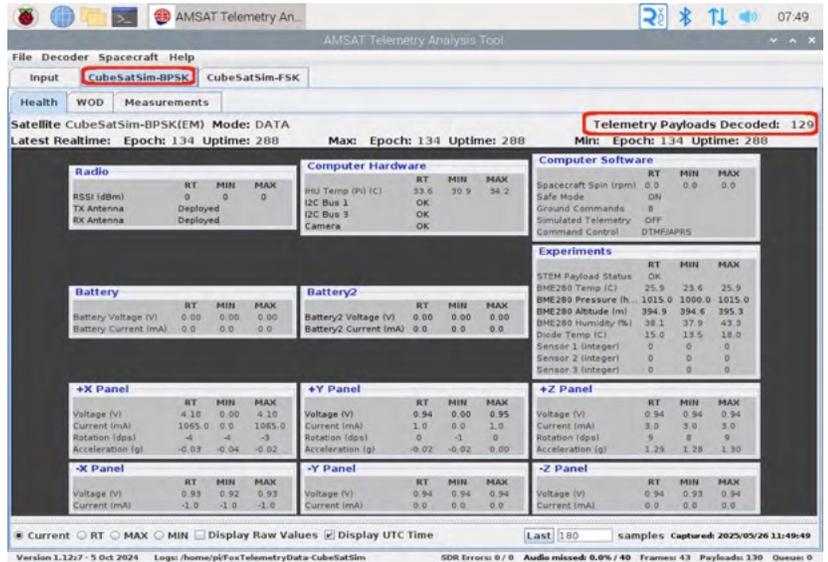
Now select the **CubeSatSim-BPSK** tab and you should be able to see that you are receiving data and decoding. Make sure that “Telemetry Payloads Decoded” in the upper right corner keeps increasing.

To Decode FSK:

Change the CubeSatSim to FSK mode (Mode 2 – two blinks of the Green LED).

Run FoxTelem and in the Input tab, set the “Telem Format” to be “FSK 200bps (Fox1)”. You might need to click on Stop before you can change it. To see data, select the **CubeSatSim-FSK** tab.

You can delete the payload files from previous sessions by clicking on Stop in the Input tab, then going under the File menu then Delete Payload Files. All the counts will go to zero and you will only see fresh data.



To Decode SSTV:

Change the CubeSatSim to SSTV mode (Mode 4 – four blinks of the Green LED).

You can use QSTTV or you can open the WebSDR app on the Raspberry Pi.

For the frequency, select CubeSatSim (yellow button) which will set the decoding the “DIG” and “SSTV”. Decoding will start when a new image transmission begins, and the CubeSat image will appear first in the image window on the left side. If you have a camera on the CubeSatSim, the second image and all of the following images will show what the camera is viewing.

To Decode APRS:

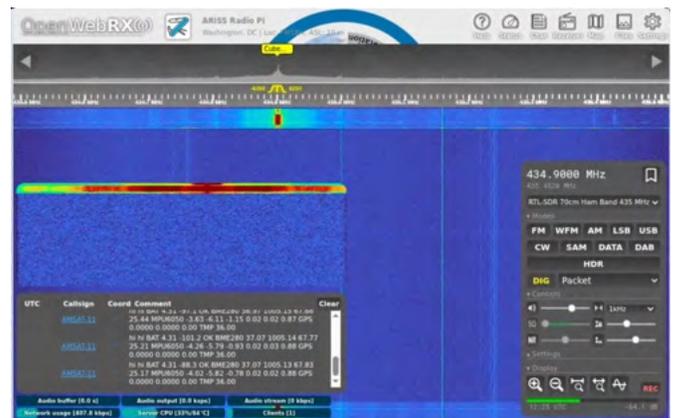
Change the CubeSatSim to APRS (Mode 1). You can run Direwolf or open the WebSDR on the desktop. WebSDR is already set up for receiving the CubeSatSim on 434.9 MHz. Make sure you select “Packet” next to “DIG” in yellow. You will see the APRS data and the signal in the waterfall display.

To Decode CW:

Open OpenWebRX and select “CW Decoder” next to “DIG”. You might need to offset the frequency from the center to get it to decode.

To Receive Cross Band Repeater:

Change the CubeSatSim to Cross Band Repeater Mode (Mode 6). Run OpenWebRX. For the band, select “RTL-SDR 2m Ham Band 145 MHz (CubeSatSim Cross Band Repeater)” and tune to 144.9 MHz FM and you should hear anything transmitted on 435 MHz (assuming default frequencies are used). See the Cross Band Repeater Quick Start Guide for more information.



Cross Band Repeater Quick Start Guide



This guide provides a short description of how to use the CubeSatSim in Cross Band Repeater mode.

What it is:

CubeSatSim v2.1 software and later has a Cross Band Repeater mode that simulates a UHF uplink, VHF downlink FM satellite repeater often known as “Mode U/V” similar to AO-91 (RadFxSat / Fox-1B) or AO-73 (FUNcube).



What You Need:

To try out the Cross Band Repeater Mode, you will need:

- CubeSatSim with v2.1 software or later.
- Two VHF/UHF FM radios (HTs), or one UHF FM radio and an SDR (such as a Baofeng and an RTL-SDR plugged into a Pi or Computer) or a VHF/UHF transceiver with full duplex (such as an Icom IC-9700)

Setting the Cross Band Repeater Mode:

There are three ways to set this mode:

- Use the pushbutton to change to the Repeater Mode 6 by pressing and holding the pushbutton and releasing when the Green Power LED blinks six times.
- Logging into your Pi Zero 2 and typing the command **CubeSatSim/config -e** (see <https://cubesatsim.org/download/cubesatsim-readme.pdf> for details of how to log into your Pi Zero 2)
- Use Command and Control (C2C) by sending **6#** with DTMF or **MODE=e** with APRS (see the Command and Control Quick Start Guide for details)

Using the Cross Band Repeater Mode:

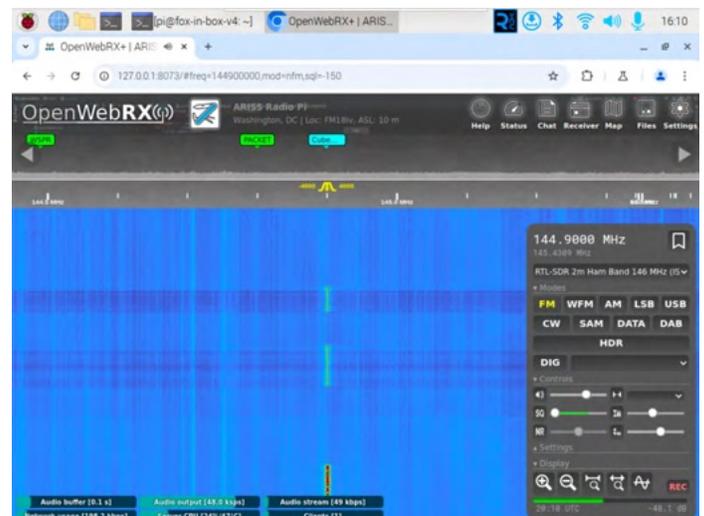
The Green Power LED will be illuminated all the time, but the Blue Transmit LED won't illuminate unless the repeater is active. No telemetry is transmitted after the initial CW ID.

The CubeSatSim will be in receive mode, listening on the default frequency of 435 MHz, or another UHF frequency set by the **CubeSatSim/config -F** command.

When the squelch is broken on that frequency, the CubeSatSim will retransmit the FM signal received at the default frequency of 144.9 MHz. Or, if another transmit frequency is set by the **CubeSatSim/config -F** command, the transmit frequency will be the set frequency minus 290 MHz (for example, the default 434.9 MHz – 290 MHz = 144.9 MHz).

The Blue Transmit LED will illuminate as long as the squelch is broken. There is about a one second timeout. If using the Raspberry Pi Ground Station, select “RTL-SDR 2m Ham Band 145 MHz (CubeSatSim Cross Band Repeater)”.

You should initially listen for the VHF repeated signal with no squelch set on your receiver as the signal will be weak. Also, there will be a small random frequency offset from expected



frequency, so you may need to tune a few kHz either side of the expected frequency – you may need to reduce your receiver’s frequency step from 10kHz or 15kHz down to 1kHz. An SDR with a waterfall is helpful to find the signal. Each time it reboots, there will be a different frequency offset, so if there is a big offset, you can try pressing and releasing the pushbutton to reboot the CubeSatSim.

Advanced Settings:

You can try different squelch level settings using the **CubeSatSim/config -q** command on the Pi Zero 2.

To prevent false triggers, you can set a CTCSS PL code on the CubeSatSim that will need to be set on the transmitting radio in order to break the squelch. For example, you could set a 67 Hz CTCSS tone just like AO-91 (RadFxSat / Fox-1B). For information see <https://github.com/alanjohnston/CubeSatSim/wiki/Command-and-Control>

To Leave Cross Band Repeater Mode:

You can change out of this mode the same way you set any mode:

- Use the pushbutton to change to another mode
- Logging into your Pi Zero 2 and using the command **CubeSatSim/config** to another mode.
- Use Command and Control (C2C) by sending the mode number then **#** with DTMF or an APRS packet with **MODE=** and the mode letter (see the Command and Control Quick Start Guide for details)

Troubleshooting:

Here are some possible problems and resolutions:

1. I get an error when I try to set the Cross Band Repeater Mode.
 - a. You may not have the CubeSatSim v2.1 software and need to update. You will need to log into your Pi Zero 2 which has a WiFi connection, then run the update command: **CubeSatSim/update** Alternatively, you can flash a new micro SD card with the latest software by following these instructions: <https://github.com/alanjohnston/CubeSatSim/wiki/2.-Software-Install>.
2. I never see the Blue Transmit LED illuminate when I transmit.
 - a. Are you transmitting an FM signal at 435 MHz? Besides the initial CW ID, you won’t hear any transmissions from the CubeSatSim unless you transmit a FM carrier on the receive frequency which is 435 MHz as the default. The Green Power LED will be illuminated all the time but the Blue Transmit LED will only illuminate when you transmit.
 - b. You can see the CubeSatSim squelch using the **CubeSatSim/config -Q** command while transmitting.
 - c. Do you have two radios (one to transmit, one to listen) or a radio and an SDR or a radio with full duplex? If so, verify you can hear or see your transmission on 435 MHz.
3. I see the Blue Transmit LED illuminate when I transmit, but I never hear anything on VHF.
 - a. Do you have two radios (one to transmit, one to listen) or a radio and an SDR or a radio with full duplex? If you only have one half-duplex radio such as a Baofeng, you can’t hear yourself while you transmit.
 - b. Are you speaking or transmitting DTMF tones? If you are just transmitting a carrier, you will hear silence.
 - c. Are you listening on the right receive frequency, 144.9 MHz by default? The signal can be shifted by a few kHz, so you might need to tune up and down a bit.
 - d. Is your squelch on your receiving radio or SDR set too high? Try turning off the squelch and turning the volume up high.
 - e. If you have a Beta v1.3.2 CubeSatSim, has it been upgraded to the v2 hardware? See the wiki for details.
4. The repeated signal is weak or garbled.
 - a. The frequency might be shifted – try going up or down a few kHz from the expected frequency or use an SDR receiver so you can see the carrier to tune accurately.
 - b. You might have your receiver squelch set too high – try turning off the squelch and turning the volume up high
5. Command and Control doesn’t work in this mode.
 - a. Is C2C turned on?
 - b. Are you transmitting a valid command?
 - c. See the Command and Control Quick Start Guide for full troubleshooting



This guide provides a short description of how to use Command and Control (C2C) with the CubeSatSim.

What it is:

Satellite Command and Control refers to the ability to connect, usually over radio, with a satellite and upload commands, settings, schedule operations and experiments, and other functions. The CubeSatSim v2 can now support radio Command and Control using a FM transmitter in the UHF band by sending either a carrier signal, DTMF tones or APRS packets to change modes or turn the transmitter on or off.



What You Need:

- A v2 CubeSatSim or an upgraded Beta v1.3.2 CubeSatSim
- A UHF FM radio or transmitter (for example, a Baofeng)
- An amateur radio (ham) license since Command and Control involves transmitting in the Ham Radio 70cm band. Before doing this, ensure you have your license and comply with all local rules and regulations. If you live in the USA, here is some information: <https://www.arrl.org/getting-licensed>

Transmit Frequency:

The default frequency which you should transmit on for Command and Control is 435 MHz. You can change this frequency by logging into your Pi Zero 2 and typing **CubeSatSim/config -F** and changing the UHF receive frequency of the CubeSatSim.

How to Turn Command and Control ON and OFF:

By default, Command and Control is OFF on the CubeSatSim. There are two ways to turn it on or off:

- Using the pushbutton. Press and hold the pushbutton until it goes through its full sequence: a series of short blinks, some longer blinks, then some even longer blinks. At this point, you can release the pushbutton. This will toggle the state. That is, if Command and Control was ON, it will turn off. If it was OFF, it will turn ON.
- Using a command line on your Pi Zero 2. You will need to log into your Pi Zero 2 (see <https://cubesatsim.org/download/cubesatsim-readme.pdf> for how to do this) then type the command **CubeSatSim/config -T** and then follow the prompts to turn it on or off.

You can tell if Command and Control is ON if the letter **C** is transmitted in CW after the callsign on power up.

Using Command and Control:

When turned on, Command and Control has three modes:

- *Carrier Command and Control.* In this mode, the CubeSatSim is just listening for a carrier signal on the receive frequency. The squelch setting on the FM Transceiver set with the **CubeSatSim/config -q** will determine the sensitivity. If a carrier is detected, the CubeSatSim will move to the next mode and the Green Power LED will blink the number of times of that mode. The sequence is APRS-> FSK-> BPSK-> SSTV-> CW-> Repeater-> FunCube-> APRS This mode is the simplest, but it can produce false triggers.
- *DTMF Command and Control.* In this mode, the CubeSatSim is listening for DTMF key presses followed by the pound (hash) sign **#**. The DTMF number corresponds to the mode number. The Green Power LED will blink with the number of the mode and then the CubeSatSim will restart or reboot. The set of commands is listed in the table.

- *APRS Command and Control.* In this mode, the CubeSatSim is waiting for an APRS packet containing the string **MODE=** followed by a letter of the new mode. The Green Power LED will blink with the number of the mode and then restart. The set of commands is listed below:

DTMF Command	APRS Command	Result
1#	MODE=a	Change to APRS Mode (mode 1)
2#	MODE=f	Change to FSK Mode (mode 2)
3#	MODE=b	Change to BPSK Mode (mode 3)
4#	MODE=s	Change to SSTV Mode (mode 4)
5#	MODE=m	Change to CW Mode (mode 5)
6#	MODE=e	Change to Repeater Mode (mode 6)
7#	MODE=j	Change to FUNcube Mode (mode 7)
10#	MODE=o	Turn transmission of telemetry on or off



When Command and Control is turned on the first time, the default mode is DTMF or APRS mode. To change it to Carrier (Squelch), log into your Raspberry Pi and enter the command **CubeSatSim/config -d**

To troubleshoot Command and Control, you can look at the Command and Control log file on the Raspberry Pi Zero 2 by typing: **CubeSatSim/log -c**

Advanced Settings:

You can change the frequency that the CubeSatSim listens on to another frequency in the UHF band by logging into your Pi Zero 2 and typing the command **CubeSatSim/log -F**

You can change the squelch level that the CubeSatSim receiver uses by logging into your Pi Zero 2 and typing the command **CubeSatSim/log -q**

The default Command and Control has no security built in. Anyone can transmit on the frequency and the CubeSatSim will respond. You can add a small amount of security by activating CTCSS PL (Private Line) codes on the FM receiver by logging into your Pi and typing **CubeSatSim/config -P** You will then need to program the same PL code into your FM transmitter.

Troubleshooting:

Here are some possible problems and resolutions:

1. I can't tell if C2C is enabled.
 - a. When you power up the CubeSatSim, if you hear a Morse Code "C" after the callsign, this means Command and Control is enabled. If you are in APRS mode the letter "C" after the telemetry string in the packet indicates that C2C is enabled. In SSTV mode, the letter C after the callsign on camera images indicates that C2C is enabled.
 - b. C2C is disabled by default in the CubeSatSim software, so you will need to turn it on using the pushbutton or by command initially.
 - c. You can log into your Pi Zero 2 and the **CubeSatSim/config** command will tell you if C2C is enabled and if DTMF/APRS (Direwolf) or Carrier (squelch) mode is set.
2. C2C is enabled, but I get no responses to my commands.
 - a. Are you transmitting FM on the receiving frequency for the CubeSatSim? It is 435 MHz by default unless changed by the **CubeSatSim/config -F** command. You can try the Cross Band Repeater Mode to see if the CubeSatSim receiver squelch is being broken your transmission – see the Cross Band Repeater Quick Start Guide. You can also monitor the squelch using the **CubeSatSim/config -Q** command.
 - b. Are you transmitting a valid command? If DTMF, you need to transmit a digit 1-7 or 10 followed by the # (pound or hash symbol). If you wait too long between sending the digit and the #, the command will fail. If using APRS, are you sending a valid APRS packet with **MODE=** and the letter **a, f, b, s, m, e, j, or o**. **MODE** must be capitalized and no spaces around the =
 - c. If you have a Beta v1.3.2 CubeSatSim, has the hardware been updated to v2? See the Wiki for details.

CubeSatSim Activity Guides

Here are some Activity Guides for the CubeSatSim that will allow you to explore the operation of the CubeSatSim. The Google Docs file can be downloaded as a Word Document or PDF by selecting `File` then `Download`, or if you have a Google Account, you can select `File` then `Make a Copy` and have your own online document.

These documents are offered with a Creative Commons By Attribution license, so you are free to edit and modify them as long as you credit their original source as AMSAT and the original author.

For each activity, there is an Activity Guide which has notes and sample data results, and a handout which could be given to students.

■ [Listening Activity Guide](#)

You can perform this activity with just the CubeSatSim and a radio or SDR to listen to the sound of the different telemetry modes. Here is a [Listening Activity](#) handout.

■ [Telemetry Activity Guide](#)

This activity takes you through using FoxTelem to view all the different types of information known as telemetry transmitted by the CubeSatSim. With a turntable and LED lamp, you can see realistic data of a CubeSat in orbit in space. Here is a [Telemetry Activity](#) handout.

■ [Advanced Telemetry Activity Guide](#)

This activity allows students to determine the CubeSatSim's status by interpreting telemetry data displayed in FoxTelem. It simulates several conditions that may occur while a cubesat is in orbit.

■ [BME Sensor Activity Guide](#)

This activity explores the BME280 Pressure, Temperature, Altitude, and Humidity sensor in the CubeSatSim. Here is a [BME Sensor Activity](#) handout.

■ [MPU Sensor Activity Guide](#)

This activity explores the MPU6050 Accelerometer and Gyro sensor in the CubeSatSim. Here is a [MPU Sensor Activity](#) handout.

■ [Solar Panel Activity Guide](#)

This activity explores the operation of the Solar Panels on the CubeSatSim. By placing the CubeSatSim in various lighting conditions such as the LED lamp, phone flashlight, and sunlight, you will see the electrical properties of the Solar Panels. Here is a [Solar Panel Activity](#) handout.

■ [Battery Activity Guide](#)

This activity explores the Nickel Metal Hydride (NiMH) rechargeable batteries in the CubeSatSim. Here is a [Battery Activity](#) handout.

■ [Troubleshooting Activity Guide](#)

This activity gives examples of how faults and failures on the CubeSatSim can be simulated and detected using the telemetry.

■ [Group Building Activity Guide](#)

This activity has suggestions on how a CubeSatSim can be built as a group activity. It describes the division of work into four parallel paths spread over six one-hour sessions.

■ [MQTT Adafruit IO Activity Guide](#)

This activity describes how to extend the CubeSatSim to support the MQTT protocol and view sensor data over WiFi. You will need to install the Arduino IDE (Integrated Development Environment) or have a Fox-in-a-Box Raspberry Ground Station which has this software pre-installed. You will need to recompile the Raspberry Pi Pico code after adding your local WiFi credentials. After setting up a free account on Adafruit.io, your sensor data will be viewable on the internet on dashboard.

AMSAT Youth Initiative Update: First Online Course Debuts at 2025 Hamvention

Frank Karnauskas, N1UW
VP, Development

Hamvention attendees had their first chance to sample the first in a series of online courses designed to interest young people in amateur satellites and amateur radio. The series is based on the theme, "Satellites in Space Help Us Live Better Lives Here on Earth."

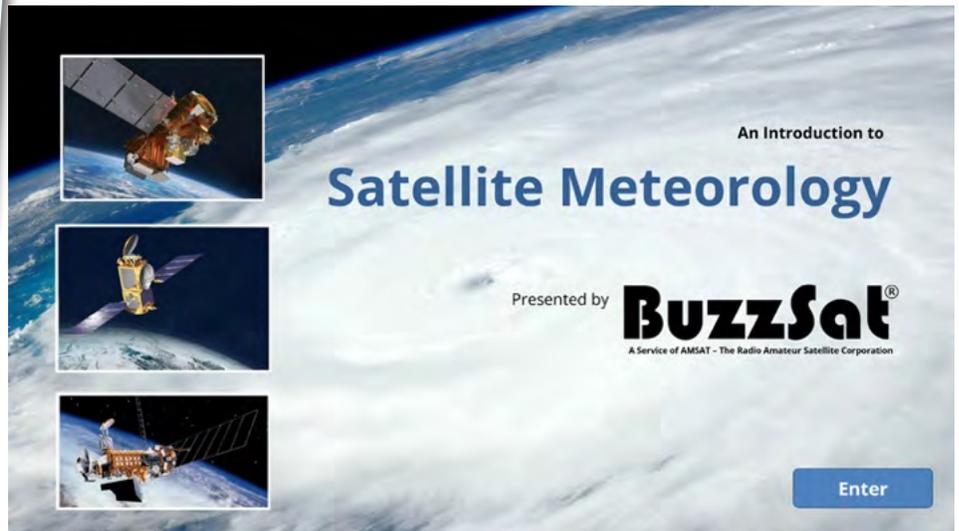
This course is the first in a series of topics currently in development, including:

- Satellite Meteorology
- Climate Change
- Pollution Control
- Preservation of Natural Resources and Wildlife
- Wildfire Detection and Firefighting
- Broadcasting
- Navigation
- Military Communications
- Robotics
- Space Exploration
- Amateur Satellites and
- Amateur Radio.

A Community-Based Program

One of the significant aspects of this program is that, although it is a STEM program, it is not school-based. The courses are free and available to all students, parents, homeschoolers, scouting leaders, and anyone who has an interest in stimulating youth's interest in satellites and communication. If teachers want to use the courses, they are welcome, of course. However, we want the material to be freely accessible to anyone who wants to use it, without requiring them to find a school system that supports it.

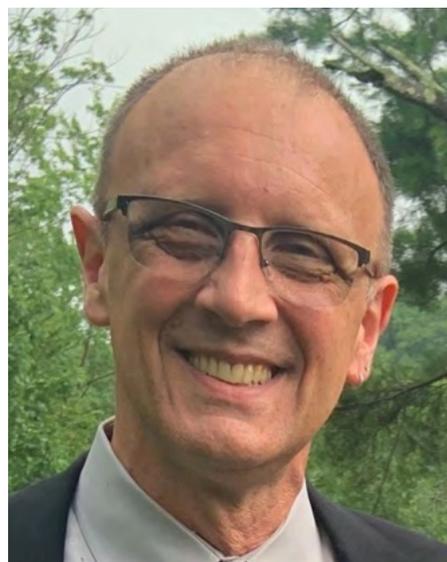
This first course, already available online at www.BuzzSat.com, is a beta release. It is currently being reviewed by professional meteorologists, educators, and in some homes. When ready, it will initially be promoted to homeschool families in one or two states. After we receive feedback and make adjustments to the program, it will be promoted nationally through various media channels.



The title page of "Introduction to Satellite Meteorology" features dramatic imagery to draw attention to the subject matter and lend it a high-tech flavor.

Volunteers Play a Key Role

The Introduction to Satellite Meteorology course was the first planned course and became a reality thanks to an AMSAT member who is a retired high school science teacher. Eric Sonnenwald, N2XSE, sent a note to the volunteer@amsat.org account explaining that he was not a rocket scientist but would like to help AMSAT in some way. He immediately got the "Have we got a deal for you!" reply, and we were off and running. Eric, a graduate of Montclair State University with an M.S. in Geosciences and the former Science Curriculum Coordinator for Hardyston Middle School, knew exactly how to get started creating our premier courseware.



Introduction to Satellite Meteorology course developer, Eric Sonnenwald, N2XSE. (Photo Credit: AMSAT.)

One of Eric's most challenging tasks in creating the lessons was sourcing copyright-free graphics to use. Fortunately, there is a relatively abundant pool of free materials available on government and college websites from which to draw.

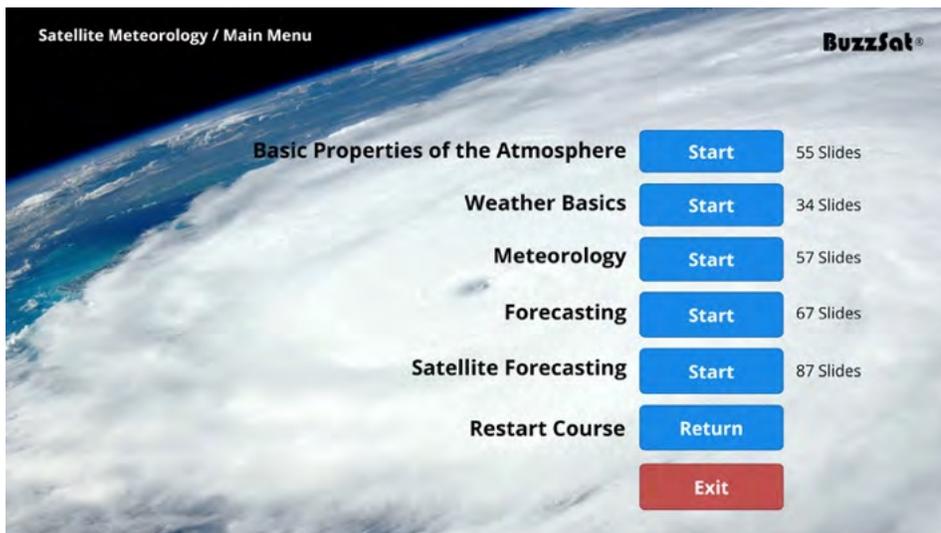
Publishing the Course

While Eric worked on creating the course outline and lesson pages, the search began for a suitable authoring system that would be used to convert the content into a system of files deliverable online. Possible solutions ranged from low-cost/low-functionality titles to very costly titles with very advanced styling, formatting, interactivity, and tracking options.

The decision was to proceed with one of the costlier options, Articulate 360. It provided a lower-cost entry point, allowing us to get started with the basics while also including a suite of tools that can be utilized as we become more experienced with the user interface.

Two possible enhancements that come with the next tier are tremendously exciting. The AI-powered translation function can convert the course content into any of 70 different languages. The text-to-speech voice synthesis utility can add voiceover narration to the courses.

With these two enhancements, the AMSAT Youth Initiative has the potential to become a worldwide destination for youth space education. We can also reach more young people with disabilities who might otherwise be unable to participate.



The list of chapters on the main menu shows the number of pages contained in each unit. At the time of its initial release, the authors had no idea how much time an average learner would require to complete it.

Keeping the Student Engaged

A common complaint of online learning systems is that they are nothing more than electronic page-turners. A student can next...next...next their way through an entire course, completing it without having learned a thing.

Our courses are designed to challenge the student at every screen - to require their full attention and to demonstrate their comprehension (or their skill at making good guesses!) before moving on.

Each page presents a concept in two or three sentences accompanied by a graphic. Each page then challenges the learner with one of three types of interactions:

- True-or-false question
- Multiple choice question
- Click on something in a picture.

If the learner responds correctly, they can proceed to the next page. If they answer incorrectly, they can try again and again until they get it right. Everyone scores 100% by the end of the course.

Next Steps

The online version of the course you see today is fully functional but is considered a beta release. The tools to enhance visual presentation with animations and other effects are plentiful, as are the abilities to track learners' activities and modify the presentation based on their past performance. These production values will be added as we become more competent with the authoring system.

However, it was essential to complete the course so that it could be showcased at Hamvention, allowing us to share the vision of what we are trying to accomplish. From that standpoint, Hamvention was a great success for the program. We received meaningful comments from educators, scouting leaders and members of specialty professions. We have also had discussions with several experts in various fields who are interested in becoming volunteer course developers.

We are confident that we are on the right track and have already begun development on our next two courses.

Solid Gold

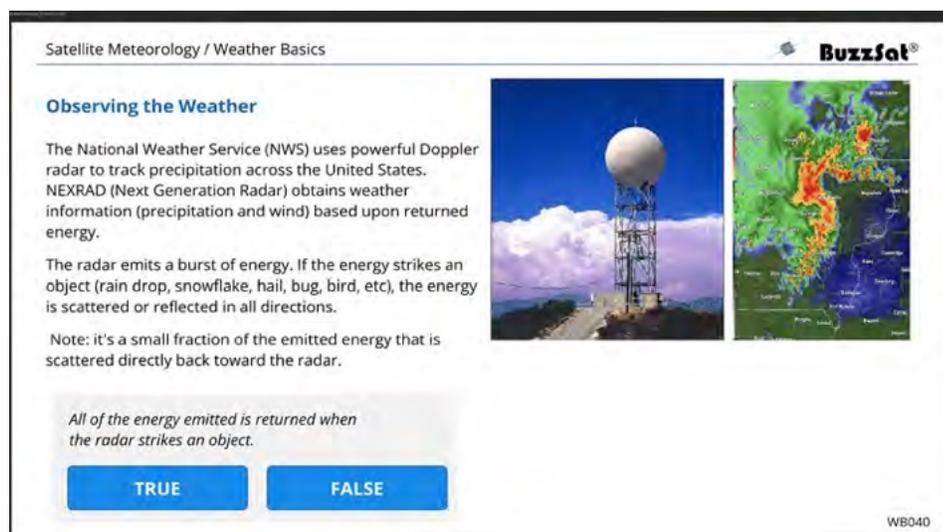
I said that Hamvention was a success because of the feedback we received from various people. But, the best feedback came at 11:10 A.M. on Sunday while we were getting ready to pack up and head home.

A father, mother and little girl, perhaps ten years old, strolled into the AMSAT booth. While the father wandered off to talk satellites with AMSAT staff, I watched the little girl, entirely on her own and without any urging from her parents, walk up to the laptops. She took a seat and began working the problem on the screen. Her mother joined her, and the two of them worked diligently on each page. If you look closely at the young girl's face, you can see she is focused on the problem. She is not wildly clicking the mouse around just for the sake of keeping busy. She is engaged.

After twenty minutes had passed, we had to start packing up. It took a bribe (a BuzzSat sticker) and the promise that she could continue on her parents' computer when she got home.

We all stop sometimes and wonder if all the volunteering we do is worth all the effort we put in. I can tell you, after watching this one little girl sitting by herself, challenging her mind, the answer is a resounding YES. What you do for AMSAT does matter!

Note: We are seeking additional subject matter experts to develop course content



To keep learners engaged, only two or three sentences are presented, focusing on a specific concept. Learners must correctly respond to the query to proceed to the next page. There is no "Next" button that allows the learner to charge through the content.

Satellite Meteorology / Weather Basics BuzzSat®

Observing the Weather

The National Oceanic and Atmospheric Administration (NOAA) collects real-time data from satellites, buoys, weather stations, citizen scientists, and more.

GOES satellites provide the kind of continuous monitoring necessary for intensive data analysis.

They circle the Earth in a geosynchronous orbit, which means they orbit the equatorial plane of the Earth at a speed matching the Earth's rotation.



GEOS satellites orbit the Earth . . .

From pole to pole .

At the same speed that the Earth rotates.

Above the equator faster than the Earth rotates.

Incorrect. Try again.

WB030

Learners can take as many times as needed to answer a question correctly. The authoring system does allow the designer to provide remedial training for incorrect responses. This function may be added in later releases of the program.

Satellite Meteorology / Satellite Forecasting BuzzSat®

Polar Orbiting Satellites

There are two basic types of satellites.

The first are polar-orbiting satellites.

They revolve around the Earth at relatively low altitudes, approximately 500 miles.

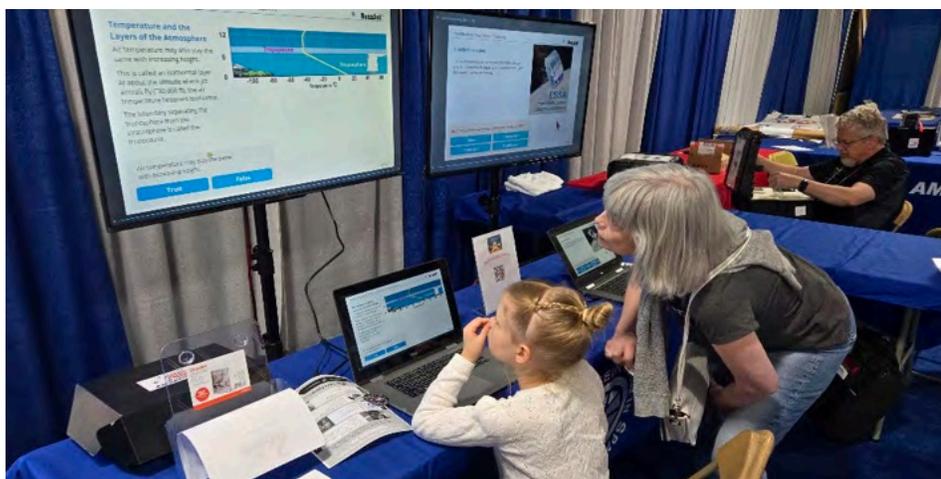
They pass over the polar regions as Earth rotates underneath.



Click on the solar panels on the NOAA Polar Orbiting Satellite.

SF070

Requiring the learner to click on a “hot spot” in an illustration is helpful when working with schematics, complex diagrams or photos of equipment.



A young visitor to the AMSAT booth locked into the course demonstration station and worked through the screens for nearly half an hour. Watching her so absorbed gave the developers the satisfaction that they were on the right track for future

for the topics listed above, as well as any other issues you believe would be suitable additions to our catalog or courses. We are also interested in speaking with developers who have experience with Articulate 360 or similar ELS systems and are willing to volunteer their time. Please contact Frank Karnauskas at f.karnauskas@amsat.org.



AMSAT Education and CubeSat Simulator Project Update

ALAN JOHNSTON, PhD, KU2Y
Vice President, Educational Relations, AMSAT
Associate Teaching Professor, Villanova University,
Electrical and Computer Engineering Department
ku2y@arrl.net

Abstract- The AMSAT CubeSatSim, the CubeSat Simulator, is a small, low cost, spacecraft simulator that serves as a tool for academic education, public demonstrations and theory and design disciplines. It can be used in a classroom, lab or training setting to introduce the basics of satellite operations, or it can be used to teach STEM (Science, Technology, Engineering, Math) exercises. This article gives an update on the CubeSatSim Project up to October 2025 since the last AMSAT Space Symposium. The update includes classroom activities, events, academic papers, and overall project status.

Overview

The AMSAT CubeSatSim model is shown in Figure 1. The AMSAT CubeSatSim, or CubeSat Simulator, is a Raspberry Pi Zero-based, 3D printed, functional model of a 1U form factor CubeSat. It is designed to act, as reasonably as possible, as 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 <https://CubeSatSim.org>. Like real LEO satellites, this simulator is self-powered through onboard rechargeable batteries and solar panels. It transmits wireless telemetry on the amateur radio UHF band in a variety of formats.



Figure 1. The AMSAT CubeSatSim

The CubeSatSim has three custom PCBs and a Raspberry Pi Zero 2 in the board stack as shown in Figure 2.

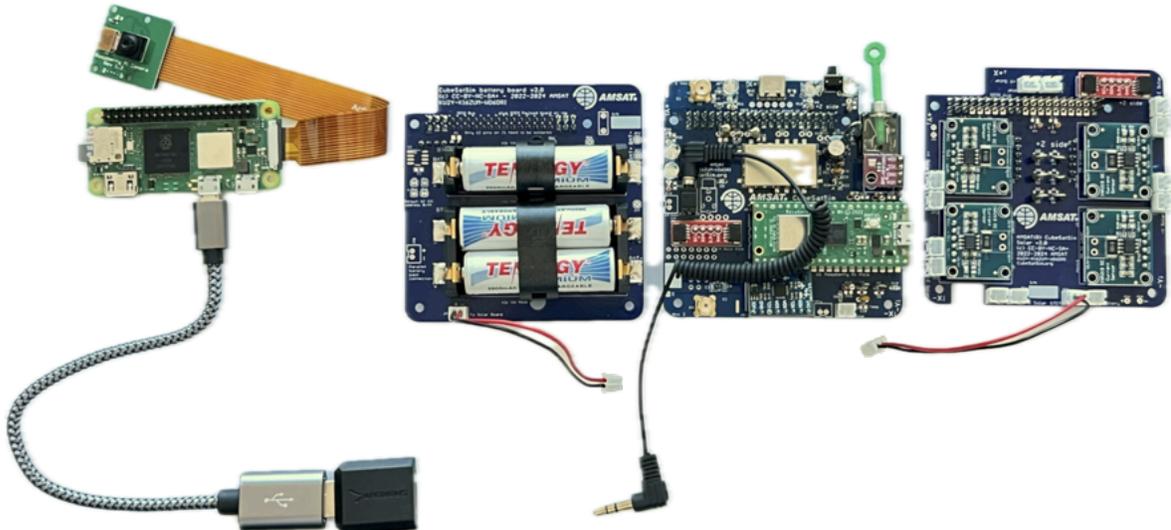


Figure 2. The CubeSatSim Beta v2 boards: From left to right, Pi Zero 2 with Pi Camera and USB sound card, Battery board, Main board, and Solar board.

The Main board is shown in Figure 3 with the major components labeled. The SR105U FM transceiver module is shown in yellow. The Raspberry Pi Pico WH microcontroller is shown in green. The BME280 Pressure/Temperature/Humidity/Altitude sensor is shown in purple. The

MPU6050 Gyro/IMU is shown in blue. The Qwiic connector for adding new sensors is shown in red.

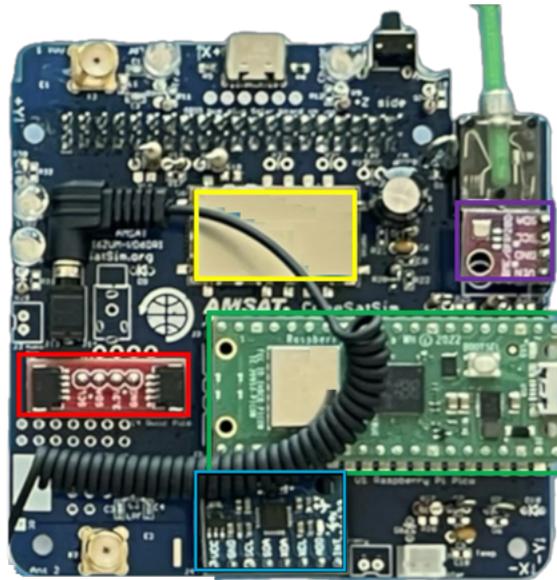


Figure 3. Main Board showing the components.

Table 1 shows the seven telemetry modes of the CubeSatSim and the decoding software used for that mode.

Blinks	Mode	Description	Decoding	Command	Audio	Waterfall(WebSDR)
1	APRS	Automatic Packet Reporting System. This digital mode sends a packet of data with AFSK or Audio Frequency Shift Keying modulation.	Windows: SoundModem or Direwolf Raspberry Pi/Linux: OpenWebRX or Direwolf	a	CubeSatSim.org/a	
2	FSK	Frequency Shift Keying. This mode transmits a continuous signal that makes a rumbling sound that emulates the AMSAT Fox CubeSats such as Fox-1C or AO-95. Also known as DUV or Data Under Voice.	Windows/Raspberry Pi/Linux: FoxTelem	f	CubeSatSim.org/f	
3	BPSK	Binary Phase Shift Keying. This mode transmits a continuous signal that sounds like noise that emulates the AMSAT Fox-1E CubeSat. You demodulate using USB.	Windows/Raspberry Pi/Linux: FoxTelem	b	CubeSatSim.org/b	
4	SSTV	Slow Scan TeleVision. This mode transmits stored images in Scottie 2 format which sounds like a series of tones.	Windows: MMSSTV Raspberry Pi/Linux: QSSTV	s	CubeSatSim.org/s	
5	CW	Continuous Wave or Morse Code. This mode transmits a FM modulated tone at 20 words per minute Morse Code telemetry.	Windows/Raspberry Pi/Linux: fdigi with spreadsheet http://cubesatsim.org/telem	m	CubeSatSim.org/m	
6	Cross Band Repeater	Simulates a U/V (UHF uplink, VHF downlink) FM satellite repeater. Default uplink is 435 MHz and downlink is 144.9 MHz.	OpenWebRX	e	CubeSatSim.org/e	
7	Fun Cube	FUNcube simulates the BPSK telemetry of the FUNcube series of CubeSats such as AO-73.	Raspberry Pi: FUNcube Telem	j	CubeSatSim.org/j	

Table 1. CubeSatSim Telemetry Modes.

The current version of the CubeSatSim PCBs is v2.0 which are available at the AMSAT Store and were also available at Hamvention. Raspberry Pi CubeSatSim SD cards and Fox-in-a-Box SD cards are also available on the store. These are 16 GB micro SD cards with the software pre-installed and configured.

There is also an active Discussion Forum on Github for the CubeSatSim which you can access with this URL: <https://github.com/alanjohnston/CubeSatSim/discussions>

The full project acknowledgements are at the end of this paper, but the key core CubeSatSim team members are:

Fredric Raab, KK6NOW
Mark Samis, KD2XS
David White, WD6DRI

Their contributions on a weekly basis to this project cannot be overstated. Without their help in fulfilling orders, testing and putting together kits, and testing and verifying every hardware and software update, this project would not be as successful as it is.

This article gives an update on the CubeSatSim Project up to October 2025 since the last AMSAT Space Symposium.

Recent Improvements

The pace of development in 2025 was similar to 2024 with several new features rolled out. The v2.1 software release added two new telemetry modes: FunCube Mode and Cross Band Repeater Mode. In addition, the Fox-in-a-Box v4 Beta Raspberry Pi Ground Station disk image was also released.

Cross Band FM Repeater Mode

The CubeSatSim Cross Band FM Repeater mode uses the FM transceiver module in receive mode, with the squelch pin read by the Raspberry Pi Zero to detect a transmission. This mode utilizes the USB sound card connected to the Pi Zero 2 with the OTG cable, and the audio jumper from the Main board to the sound card provides the received audio to the Pi Zero. The Pi Zero 2 then connects the microphone input to the rpitx transmitter. One of the CubeSatSim antennas is receiving the UHF signal, the other is transmitting the VHF signal at the same time, just like a real CubeSat with an FM repeater.

The v2.0 software release had a Repeater mode, but you couldn't turn it on and off with the pushbutton, and it was unstable in that the repeater function would stop working and require a

reboot. It also had uplink and downlink frequencies in the UHF band which made reception difficult.

The v2.1 software release has fixed these issues, and moves the downlink to the VHF band. This provides a “U/V” operation with an FM UHF uplink, VHF downlink similar to many AMSAT CubeSats such as AO-91 (RadFxSat / Fox-1B) and AMSAT-UK AO-73 (FunCube).

To try out the Cross Band Repeater Mode, you will need two VHF/UHF FM radios (HTs), or one UHF FM radio and an SDR (such as a Baofeng and an RTL-SDR plugged into a Pi or Computer) or a VHF/UHF transceiver with full duplex (such as an Icom IC-9700). These options are shown in Figure 4.



Figure 4. Radio Options for the Cross Band Repeater Mode: Two HTs, One HT and one SDR, or One Full Duplex Transceiver

In the Cross Band Repeater mode, the CubeSatSim is listening on the default frequency of 435 MHz. When the squelch is broken on that frequency, the CubeSatSim will retransmit the FM signal received at the default frequency of 144.9 MHz. Figure 5 shows the Raspberry Pi Ground Station receiving the VHF downlink signal.

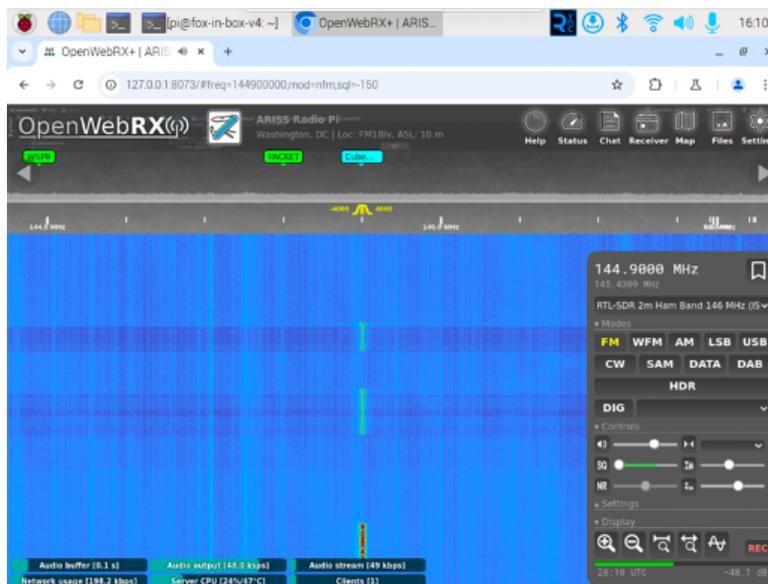


Figure 5. Raspberry Pi Ground Station based on the Fox-in-a-Box image for monitoring the VHF downlink using the OpenWebRX WebSDR.

New FunCube Mode

The CubeSatSim now has a new FunCube Mode 7. This BPSK (Binary Phase Shift Keying) 1200 bps (bits-per-second) transmission mode emulates the AMSAT-UK FunCube CubeSat (AO-73) and several other CubeSats that also use this transmission including Nayif 1 (EO-88), UKube-1, and JY1 Sat (JO-97). The transmission is very similar to the Fox-1E BPSK 1200 bps transmission already supported by the CubeSatSim in BPSK Mode 3. However, it has a different data structure and error correction. Special thanks to Duncan Hills, the lead FunCube Dashboard developer from AMSAT-UK who provided technical details and support to get this code working!

If you listen to this mode, it sounds similar to BPSK, except for every four seconds it sounds like there is a “beep” or tone.

To decode FunCube telemetry, you need a FunCube Dongle (FCDPro or FCDPro+) and decoding software (an RTL-SDR dongle will not work). For the FunCube satellites in orbit, there is a Windows Dashboard for each satellite that decodes the telemetry and uploads the data to the AMSAT-UK Data Warehouse. We don't yet have a Windows Dashboard, but we do have a Raspberry Pi application which will decode it, shown in Figure 6. It is part of the new Fox-in-a-Box-v4 beta image.

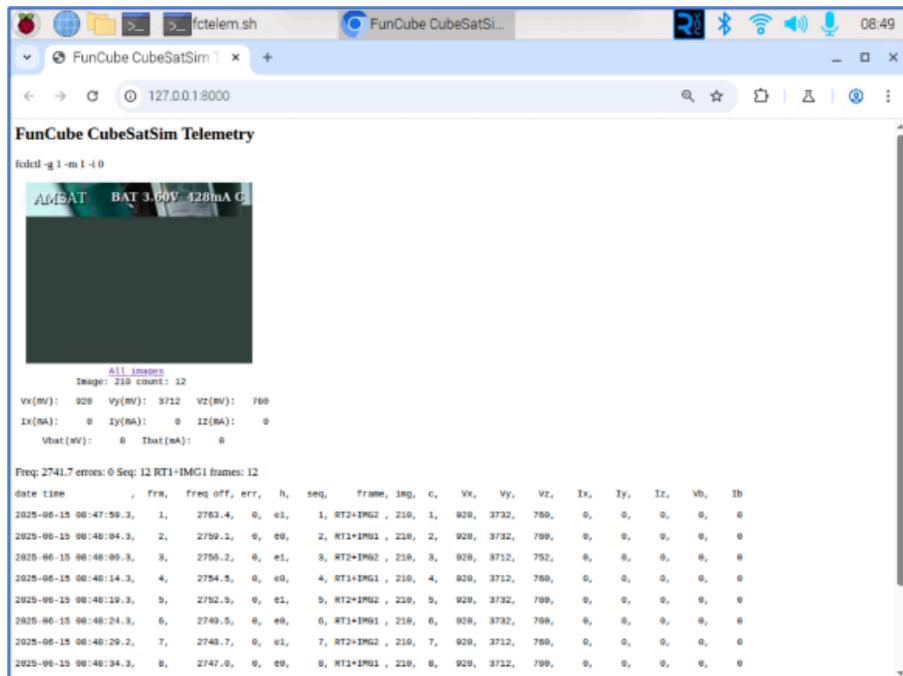


Figure 6. FunCube Telemetry Decoding using a FunCube Dongle on a Raspberry Pi with FIABv4

The telemetry data is a similar format to JY1 Sat (JO-97) with a camera image and telemetry data in each frame.

Unlike the CubeSatSim SSTV mode 4 which uses tones to transmit image pixels in Scottie 2 format, FunCube uses SSDV to encode JPEG images suitable for packetization. It uses the Reed-Solomon codec written by Phil Karn, KA9Q. You can see the image come in block by block, and if a frame is lost, there is a rectangular dropout in the image. Otherwise, the image quality is perfect. This makes an interesting comparison with the analog SSTV Mode 4 in which the image quality degrades as the signal strength decreases.

Fox-in-a-Box v4 Beta Ground Station Software

The Fox-in-a-Box (FIAB) Raspberry Pi disk image has been available from AMSAT for a long time. It has always provided a Raspberry Pi disk image with FoxTelem pre-installed for receiving telemetry from AMSAT's CubeSats, starting with Fox-1A (AO-85) through MESAT1 (AO-122). FIAB runs on a Raspberry Pi with a FunCube or RTL-SDR v3 dongle and an antenna, and can be used to receive, decode, and upload telemetry.

Starting with the FIABv3, the ARISS Radio Pi project was merged with FIAB which not only provided FoxTelem but other satellite and SDR software including:

- Direwolf for APRS decoding
- QSSTV for SSTV decoding
- CubicSDR for general SDR operation
- OpenWebRX as a WebSDR, a minimal configuration web-based SDR interface
- Gpredict for tracking satellites
- RTL-TCP for connecting to an RTL-SDR dongle over the internet
- KLAtracker for monitoring satellite passes
- PacSat a ground station for Packet Satellites

The new FIABv4 Beta image, shown in Figure 7, has all the capabilities of FIABv3 in addition to:

- Improved simulated telemetry providing more realistic data.
- Support for the Raspberry Pi 5
- Support for the RTL-SDR v4 dongle (except in FoxTelem which does not support it yet)
- OpenWebRX+ which adds built in decoders for SSTV and CW modes
- FunCube Telem, a new experimental telemetry decoder for FunCube mode
- SDR++ an open source SDR which has the same user interface in Windows and Mac as on the Pi

As before, the FIABv4 image is available for download so that you can write it to a 16GB or larger micro SD card. <https://cubesatsim.org/download/fiabv4-beta.iso.zip> It is also available for purchase on the AMSAT Store.

The FIABv4 Raspberry Pi Ground Station software is recommended for the CubeSatSim as it has the FunCube Mode 7 telemetry decoder.

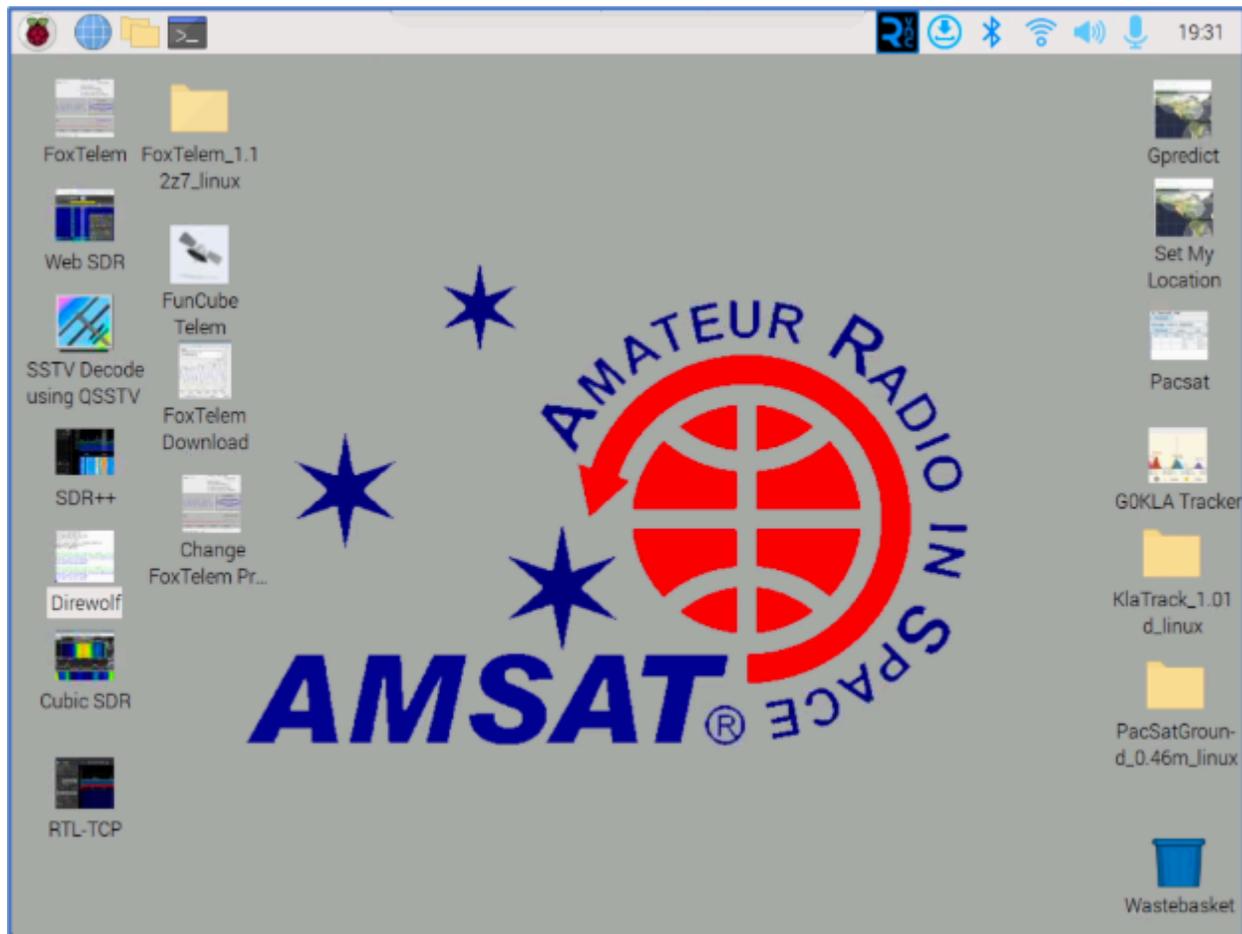


Figure 7. Fox-in-a-Box v4 Beta Raspberry Pi Image

Future Updates

We are also working on a software update for v2.2 which will add these capabilities:

- Improved simulated telemetry providing more realistic data.
- Support for BME and MPU sensors attached to Pi Zero 2.
- Simulated failures.

The ability to simulate 10 different failures which can be detected in the telemetry is a nice new feature.

We are also working on some v2.1 hardware updates. These updates will improve battery charging and will add a PTC resettable fuse to the Main board.

Lite Board

The CubeSatSim Lite board was last available as a beta board in 2022. The new Lite v2 board under development is shown in Figure 8.



Figure 8. CubeSatSim Lite v2 Board.

The Lite board is a minimal board based on the CubeSatSim Main board that plugs into a Pi Zero or Pi Zero 2 and generates all seven telemetry modes with simulated telemetry. It can also be run on a battery, but does not have a built-in charging circuit. It does not have a Raspberry Pi Pico microcontroller, but using the Qwiic connector, a BME and/or MPU sensor can be added, along with a GPS module. With the addition of a USB sound card connected to the Pi Zero with an OTG cable and an audio jumper wire, it can support Command and Control (C2C). Finally, if the RF attenuator is disconnected, the 500 mW FM transmitter can be used as a high altitude balloon payload.

Look for the Lite board to be in the AMSAT store in early 2026.

Quick Start Guides

Besides the basic CubeSatSim Quick Start Guide, there are now three other single page guides. They are shown in Figure 9 and include Ground Station, Cross Band Repeater, and Command and Control. The Quick Start Guides are available at <https://CubeSatSim.org/qsg>

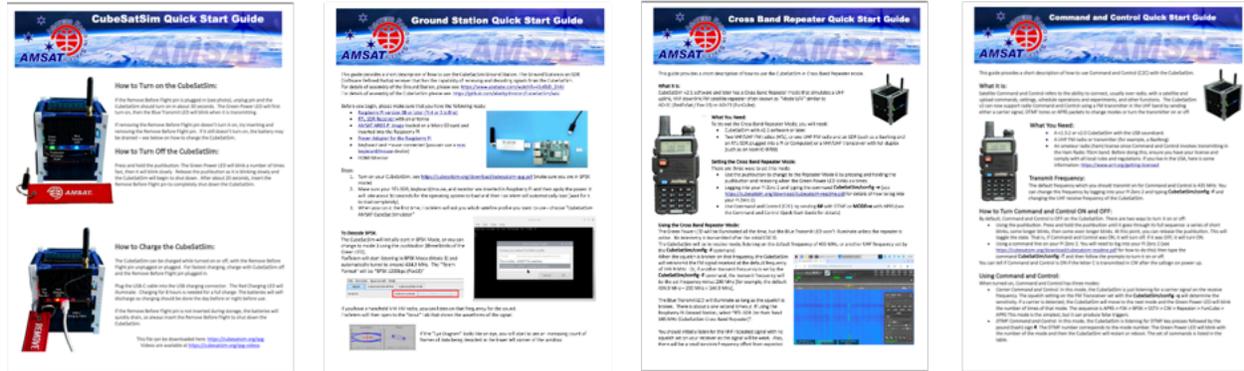


Figure 9. CubeSat Sim Quick Start Guides including Ground Station, Cross Band Repeater, and Command and Control.

Activity Guides

In 2025, we released the first set of CubeSatSim Activity Guides.

The problem statement for the Activity Guides is as follows:

I just built (or borrowed) a CubeSatSim. What can I do with it?

Activity Guides are step-by-step instructions for how to do a STEM educational activity using a CubeSatSim. Most work with just a CubeSatSim (or even just a Lite). Some are very easy to do, such as listening to the sounds of the different modes on a HT and/or looking at modes on an SDR waterfall. Others are more challenging such as doing telemetry analysis using FoxTelem.

Activity Guides include questions for students to answer as they do the activity. Each also has an Instructors Guide which has example data and answers to the questions. The CubeSatSim Activity Guides are released under a Creative Commons with Attribution license so anyone can edit and modify them as long as they indicate they were originally developed by AMSAT. This allows teachers to customize them and include them in their lesson plans.

They have been under review over the summer - thank you to our reviewers who have provided feedback! The initial set of ten are published on GitHub now

<https://github.com/alanjohnston/CubeSatSim/wiki/CubeSatSim-Activity-Guides> More will be available in the future, and we plan to record videos as well.

Here is the list of the current set of Activity Guides:

- Listening Activity Guide
- Telemetry Activity Guide
- Advanced Telemetry Activity Guide
- BME Sensor Activity Guide
- MPU Sensor Activity Guide
- Solar Panel Activity Guide
- Battery Activity Guide
- Troubleshooting Activity Guide
- Group Building Activity Guide
- MQTT Adafruit IO Activity Guide

Figure 10 shows the first page of the MPU Sensor Activity Guide.

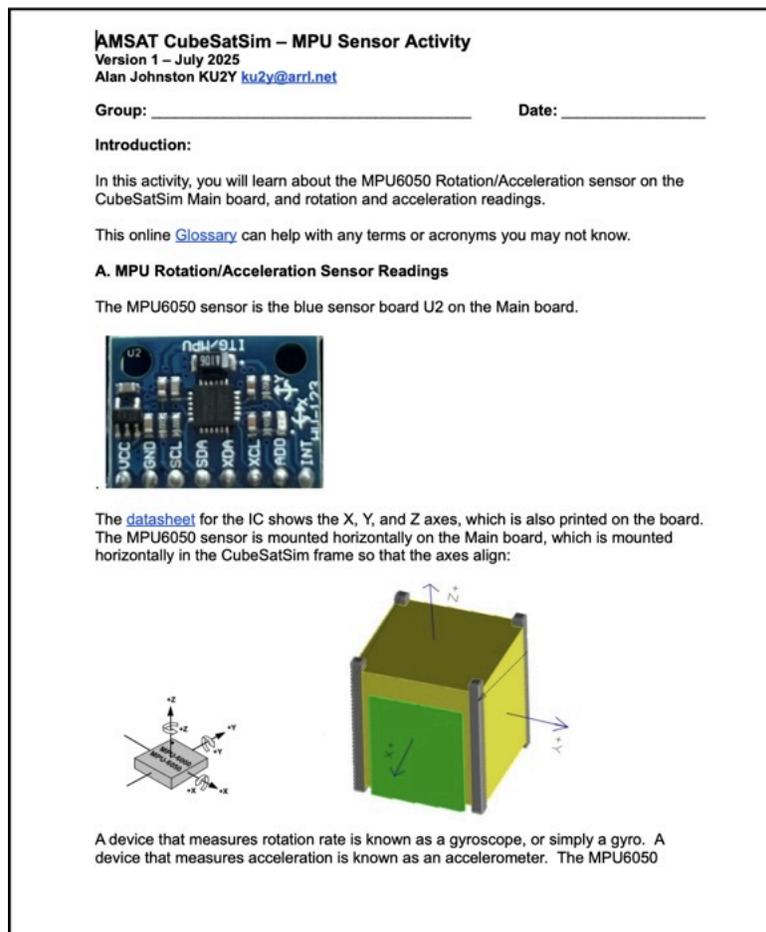


Figure 10. MPU Sensor Activity Guide.

If you have a CubeSatSim, why not give one a try and let us know how it goes.

Educational Outreach

Recent AMSAT educational outreach activities include demonstrations at Hamvention, hamfests, and other events. Loaners and Kits continue to be key to its use in STEM outreach. The CubeSatSim has also been used in college level classes as part of engineering courses. For the first time, it has been used at the ARRL Teachers Institute. Finally, there have been several academic papers and conference presentations about the CubeSatSim.

CubeSatSim Loaners

The CubeSatSim Loaners have continued to be popular throughout the year. A Loaner is a fully assembled CubeSatSim in a hard shipping case with a Raspberry Pi touch screen ground station, LED lamp, and turntable. AMSAT members can borrow a loaner to show at an event or in a classroom. Any teacher can also borrow a loaner – just contact me via ku2y@arrl.net.

CubeSatSim Kits

We also have kits available periodically. So far, 100 kits have been produced. The kit is shown in Figure 11. The kits have all three PCBs fully assembled. No soldering or 3D printing is required. After a few hours of assembly, you will have a working CubeSatSim! This year, some were sold at Hamvention and some on the AMSAT Store. Another batch will be available on the AMSAT Store in the December/January time frame.



Figure 11. CubeSatSim Kit Parts.

We now have two Loaner Kits available thanks to a grant from Quarter Century Wireless Association (QCWA). Each is similar to a Loaner in that it has a Pi Ground Station, turntable, and LED lamp in a hard shipping case. However, the CubeSatSim is in parts as a no-solder kit. It requires an hour or two of assembly to put together the CubeSatSim. The Loaner Kit is returned assembled, then we disassemble it ready for the next loan!

A Loaner Kit was recently trialed by students of the San Diego area Mount Carmel High School Amateur Radio Club (MCHSARC). The club sponsor is teacher John Earnest, KG6EQU. Assembly and demonstration of the CubeSatSim was very successful. Figure 12 shows the students assembling it:

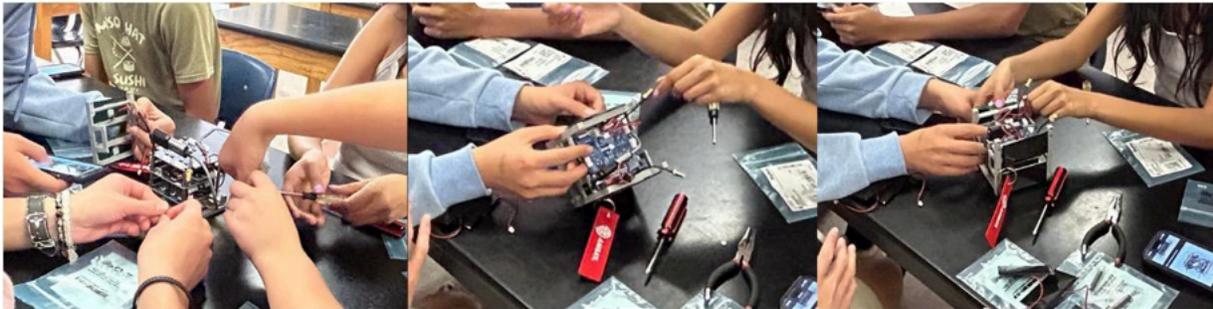


Figure 12. Students of the Mount Carmel High School Amateur Radio Club (MCHSARC) W6SUN radio club building the CubeSatSim Kit Loaner. Photo courtesy of David White, WD6DRI.

The AMSAT CubeSatSim Loaner Kits are available to anyone who wants to take into a classroom or build as a group.

ARRL Teachers Institute

For the first time in 2025, the ARRL Teacher's Institute used CubeSatSim kits in their Space Comms and Radio Astronomy summer class:

<https://www.arrrl.org/teachers-institute-on-wireless-technology>

The CubeSatSim team worked closely with Wayne Greene, KB4DSF, and Steve Goodgame, K5ATA, in the spring to put 20 kits together for them to use in two summer sessions of the class. Figures 13 and 14 show some photos from the event.

They spent almost a full day of the weeklong class building and testing their own CubeSatSim. Their kits were very similar to the kits sold on the AMSAT Store except the BME and MPU sensors needed to have pin header soldered on. This provided a review of soldering that they had done in an earlier Teachers Institute class. At the end of the week, they then took the assembled CubeSatSim home, and hopefully will utilize it in their classroom soon!



Figure 13. Photo from ARRL Teachers Institute Space Comms and Radio Astronomy class using CubeSatSim Kits. Photo courtesy of ARRL.



Figure 14. Photo from ARRL Teachers Institute Space Comms and Radio Astronomy class using CubeSatSim Kits. Photo courtesy of ARRL.

ARISS STAR Project

The CubeSatSim is a part of the ARISS (Amateur Radio on the ISS) STAR (Space Telerobotics with Amateur Radio) Project. In SSTV mode with a Pi Camera, a CubeSatSim Lite mounted on the XRP Experiential SparkFun robot rover provides images, as shown in Figure 15. The XRP rover is controlled from an ARISS Pi Ground Station with a new STAR app which now hosts the SSTV images broadcast from the rover. The XRP rover can receive commands sent via APRS or from other locations on the Web via MQTT that are passed to the STARApp's Mission Controller.

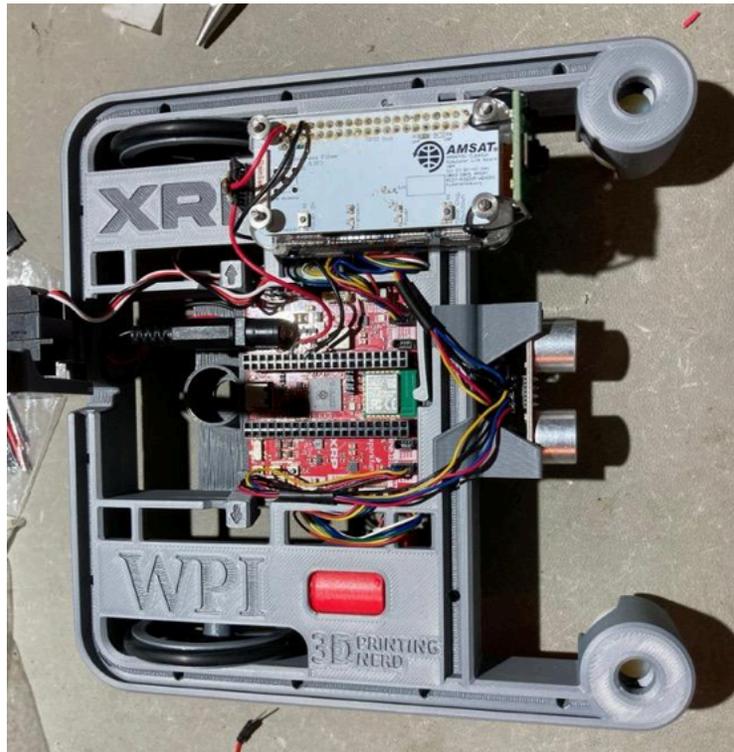


Figure 15. ARISS STAR Project XRP Robot Rover with CubeSatSim Lite with Pi Camera.

CubeSatSim in the Classroom

The CubeSatSim continues to be used in classes at a number of universities and colleges.

At Villanova University, Villanova, PA, students in the Electrical and Computer Engineering Freshman Projects class taught by Alan Johnston, KU2Y and Xun Jao, built and tested 5 CubeSatSims. Four of them are shown in Figure 16.

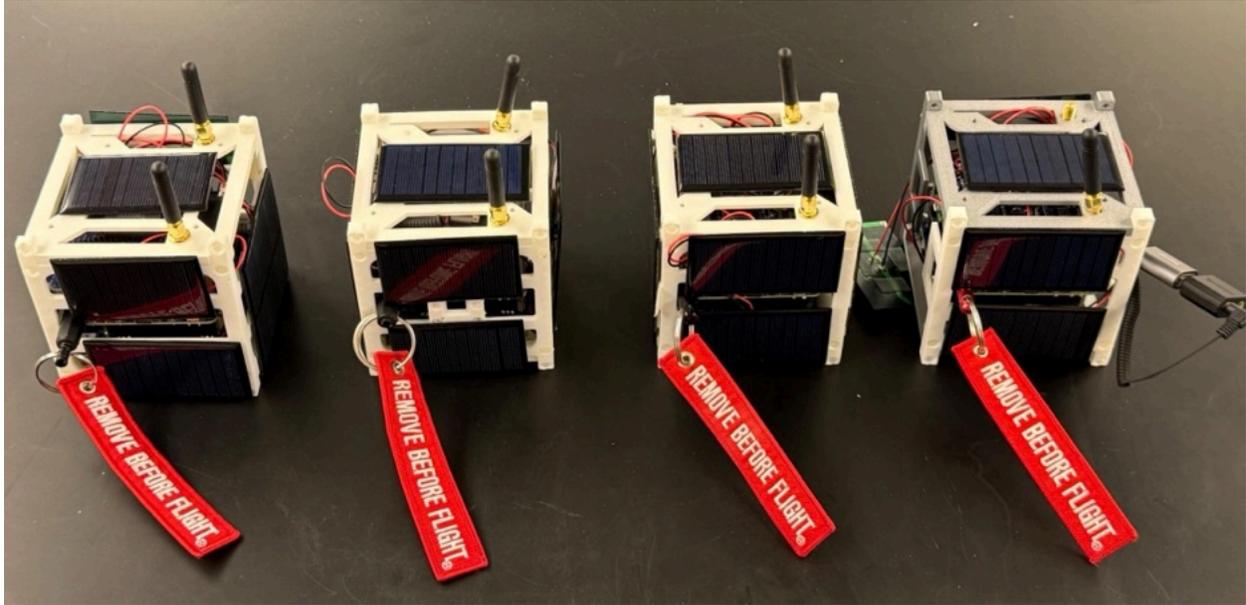


Figure 16. Four CubeSatSims built by Villanova University students in a Freshman Electrical and Computer Engineering Class.

At Johns Hopkins University, Baltimore, MD, Drew Knuth continues to use CubeSatSims and Lites in a Small Satellite Development and Experimentation capstone graduate class. A conference paper on the topic was presented this year - see the CubeSatSim in Academia section below.

At the University of Texas at Dallas, TX, Ron Dang, K5SUS, Kangkook Jee, Jaehyun Park, and Agustya Bose organized another satellite workshop at University of Texas at Dallas where they built and tested CubeSatSims. Here is the workshop website: <https://satworkshop.syssec.org/> See Agustya Bose's symposium paper for more details about this excellent event.

At Iowa State University, Ames, IA Matthew Nelson is planning to build CubeSatSims in their Make to Innovate (M2I) and Cy Sat program.

CubeSatSim at Events

CubeSatSims were utilized at many events in 2025.

In February, the CubeSatSim was shown at Hamcation in Orlando, FL by Dennis Veselka, KI4KNC. Figure 17 shows his excellent display including live FoxTelem on a monitor and multiple CubeSatSims and PCBs.



Figure 17. Hamcation CubeSatSim display showing v1.2 CubeSatSim, v2 CubeSatSim, and blank and soldered PCBs under plexiglass. Photo courtesy of Dennis Veselka, KI4KNC.

In March, a CubeSatSim was displayed at an AMSAT table at the Cheanago Valley Amateur Radio Association Hamfest organized by Don Rhodes, KB2YSI, in New York state.

In April, AMSAT Ambassador Philip Jenkins, N4HF, borrowed a CubeSatSim loaner and showed it at RARSfest, the Hamfest by the Raleigh Amateur Radio Society in Raleigh, NC.

In May, Jack Spitznagel, KD4IZ, Carsten Glasbrenner, KQ4SJM, and Dennis Veselka, KI4KNC, ran the education table at the AMSAT booth and displayed the CubeSatSim and Pi Ground Station.

In June, Tom Schuessler, N5HYP, and his team again ran an AMSAT table at the Moon Day event at the Frontiers of Flight Museum in Dallas, TX. He used a CubeSatSim loaner for an SSTV selfie display.

For ARRL Field Day in June, Rusty Moore, K1FVK, borrowed a CubeSatSim loaner and used it during the day in North Carolina.

In August, John Bews, VK4JBE, showed off his v1.2 CubeSatSim at the “Switch on to Science 2025” event at the Redlands Coast Museum in Brisbane, Queensland, Australia. The table was organized by Bayside and Districts Amateur Radio Society (BDARS).

In October, the AMSAT-UK Colloquium in October will also feature a CubeSatSim demo by Patrick Wood, 2E0IFB.

Garth Likens and Nick Pugh, K5QXJ, showed off a CubeSatSim loaner at multiple events for their CAPE Satellite Program at the University of Louisiana at Lafayette.

CubeSatSims in Academia

Greg Magone received his PhD from Colorado State University, Ft Collins, CO this year with his dissertation “Scalable system architecture for CubeSat test & evaluation for enhanced mission success”. He borrowed a CubeSatSim Loaner during his research and used it as a model to do tests. His dissertation can be downloaded here:

<https://mountainscholar.org/items/52206ea7-01d7-4eea-9bfd-859fe61edc15>

At the Universidad de Navarra, Pamplona, Spain, Juan Lacosta Arpide received his Degree in Telecommunications Systems Engineering with his final thesis project titled "Analysis of a satellite platform based on AMSAT V.1.3.2 for the study of sustainable mobility." He carefully analyzed all aspects of the design of the CubeSatSim Beta v1.3.2, even performing measurements of the range of its transmitters. The thesis can be downloaded here:

<https://dadun.unav.edu/entities/publication/27b99943-dddf-429e-81f8-3b1b6c952182>

Luke Korth, James Gardiner, and Miranda Threewitt published a paper “Building a CubeSat Capstone for Master's Students” at the 2025 IEEE Aerospace Conference in Big Sky, MT. Their paper describes how they redesigned the Johns Hopkins Space Systems Engineering master's program capstone course by its graduating students around the CubeSatSim. The paper is available from the IEEE or from the authors

<https://ieeexplore.ieee.org/abstract/document/11068771>.

Participating in the CubeSatSim Project

You can participate in the AMSAT CubeSatSim Project! Here are some of the ways:

- Document your CubeSatSimulator build and testing on social media
- Offer to demonstrate your CubeSat Simulator to local schools, your ham club, Makerspace or Hackerspace gatherings, or other STEM event

Conclusion

This paper provides an update of the previous twelve months for AMSAT Educational activities and the CubeSatSim Project. Your suggestions and comments on ways to improve the AMSAT CubeSat Simulator project are always most welcome!

Acknowledgements

We would like to thank all the CubeSatSim Project team volunteers:

Jim McLaughlin, KI6ZUM
David White, WD6DRI
Paul Graveline, K1YUB (SK)
Fredric Raab, KK6NOW
Mark Samis, KD2XS
Sopwith, N1SPW
Kerry Bonin, KJ7HTG
Jim Nagle, KF4OD
Virginia Smith, NV5F
Chris Thompson, G0KLA/VE2TCP
Christine Mehner, MD, PhD, KO4EWG
Kai Ji, AC3EN
Melissa Pore, KM4CZN
Lindsay White, KI6LZN
Randy Standke, KQ6RS

Thanks to Mark Spencer, WA8SME, for his trailblazing work on CubeSat simulators and to Bob Bruninga, WB4APR, for ideas and inspiration from his undergrad "LabSat" developments.

Pat Kilroy, N8PK, was instrumental in getting the CubeSat Simulator project going again.

We would also like to acknowledge all the open source hardware and software that is a part of the AMSAT CubeSatSim.

Finally, we would like to acknowledge the support of the AMSAT Board of Directors and the members of AMSAT for their support and encouragement of this project.

Bridging Orbit and Classroom: SatNOGS and CubeSatSim Activities

Fredric Raab KK6NOW

Adjunct Faculty, College of the Desert, Palm Desert, CA, US
fraab@collegeofthedesert.edu
fredricraab@yahoo.com

Abstract:

This paper addresses the challenge of interpreting satellite telemetry data and relating it to real-world satellite behavior. It introduces a powerful synergy between SatNOGS, a global open-source network offering free access to over 12 million satellite data downloads, and the CubeSatSim (CSS), AMSAT's low-cost satellite simulator.

SatNOGS enables students to "adopt" on-orbit CubeSats, providing the experience of being a satellite operator by scheduling downlinks and monitoring dashboards. To overcome the difficulty in interpreting this real-world telemetry, the CubeSatSim offers a hands-on, controlled environment. Students manipulate the CubeSatSim to directly understand how sensor data reflects solar panel performance, orientation, spin rates, and radio communications. This practical experience enhances their ability to interpret SatNOGS dashboards, demystifying the conversion of "waterfall signals into bits" and "bits into data". Together, SatNOGS and CubeSatSim provide both real-world data and practical understanding of telemetry data.

Introduction:

Every year, dozens of educational institutions place CubeSats into orbit. Many more institutions and students aspire to join their ranks but cannot due to costs and limited launch availability. However, these students can gain the knowledge and experience of building and operating a CubeSat by assembling AMSAT's CubeSat Simulator and by "adopting" an existing on-orbit CubeSat on SatNOGS.

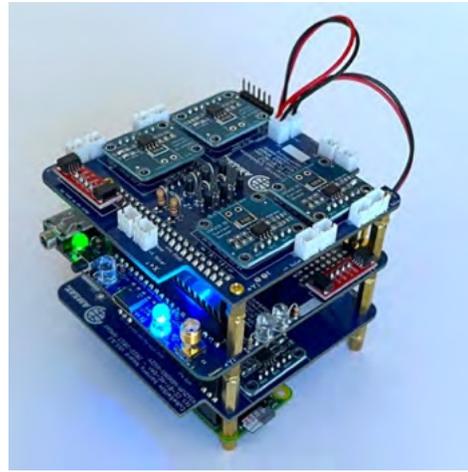
AMSAT's CubeSat Simulator (CSS) is an inexpensive, fully functional CubeSat, but is not space-worthy. It contains most of the components of a 1U CubeSat: frame with solar panels on each side, rechargeable batteries, power monitoring, flight computer, UHF/VHF radio board, camera, payload computer, inertial measurement unit (IMU) and environmental sensors. It is available as a no-solder kit that can be assembled by students in less than a day.

The CSS transmits operational telemetry data typically sent by all CubeSats: power output (voltage and current) of each solar panel, battery power, IMU data, sensor data, general health of each subsystem. This data is received using an inexpensive Software Defined Radio (SDR) dongle connected to a Raspberry Pi or a Windows system. The raw data is interpreted in real time and displayed by an AMSAT developed program: FoxTelem.

AMSAT has developed a series of hands-on educational activities for the CSS and FoxTelem to teach students how to read telemetry data and relate it to satellite performance.



Assembled CubeSatSim



Internal 4-board stack

SatNOGS is a global network of open-source ground stations operated by educational institutions, hobbyists and space enthusiasts. These stations receive data from orbiting educational satellites and upload the data to an open access database that as of September 2025 contained more than 12 million observations from 2300+ satellites. This data is available to anyone via the SatNOGS website. In addition to raw and decoded data, dashboards have been developed for some satellites to provide visual access to operational data.

Much of the operational data contained within SatNOGS is the same as reported by the CSS. And SatNOGS dashboards are similar in function to FoxTelem graphs.



A minimal SatNOGS ground station, consisting of a Raspberry Pi, SDR dongle and dipole antenna, can be built for less than \$150US. Software and detailed instructions are available on the wiki found on the SatNOGS website.

Ideally the antenna is outdoors and elevated, but as this station shows, a functional station can be indoors as long as it has a clear view of some portion of the sky. Once the station is on-line and registered, students can schedule observations (downloads) on multiple SatNOGS stations around the globe.

By selecting a dozen geographically spaced stations, they can establish their own global virtual down-link network.

Adopt a Satellite:

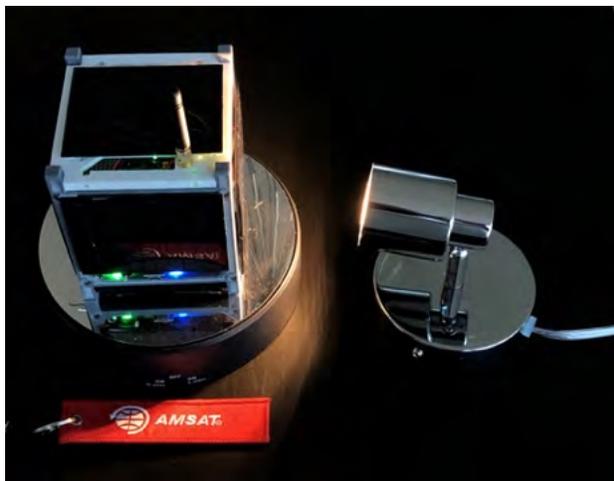
Once a CubeSat is on-orbit, its operators need to continuously monitor its health and collect scientific data. By “adopting” an existing, on-orbit CubeSat, students have the experience of being a satellite operator without the expense and risk of launching their own CubeSat. Using published information, students learn about their “adopted” CubeSat, its mission and orbital path, communications protocols used and data returned, etc. By using SatNOGS to schedule data downlinks across their “virtual network” and monitoring the CubeSat’s dashboard over a 1 to 4 week period, students learn about the CubeSat’s communications, operations and performance. As well as experiencing some of the activities of a satellite operator.

Using the CSS to Understand SatNOGS data:

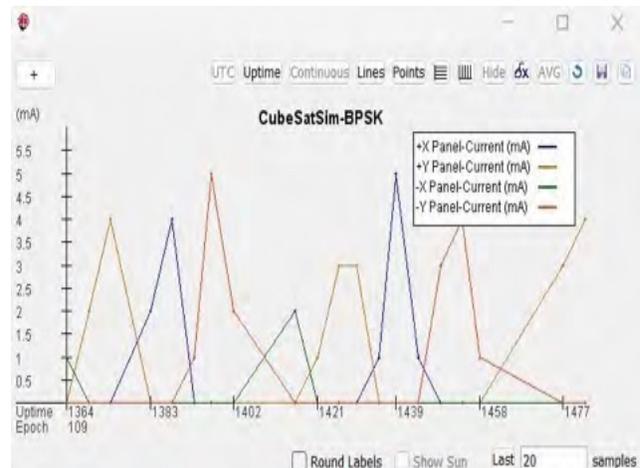
One challenge from viewing SatNOGS telemetry is relating it to the satellite’s orientation and behavior in space. It’s a dynamic environment of constantly changing data. It may be difficult to interpret data abnormalities.

The CSS provides a controlled, hands-on environment to collect data from a CubeSat. The CSS can be held stationary for an extended period of time allowing students to note readings from the solar panels and IMU. Then it can be rotated 90 degrees showing students how the readings change with position and orientation. After repeating this exercise with multiple positions on each of the three axes, students can construct a mental image of how telemetry data relates to satellite orientation.

Students can take the CSS in hand and simulate a tumbling CubeSat. Placing the CSS on a rotating turntable in front of a bright light source (or better in outside sunlight), students can use FoxTelem to plot how the solar panels output change with each rotation. And by slowly moving a cardboard panel between the CSS and light source, students can simulate the CSS entering and leaving eclipse.



CSS on turntable



Resulting FoxTelem plot

The CSS software also has a simulated telemetry mode in which the CSS transmits nominal data along with data indicative of a failed component. Students are challenged to determine the cause of the failure.

The knowledge gained from these CSS activities enable students to better understand the telemetry from their “adopted” satellite.

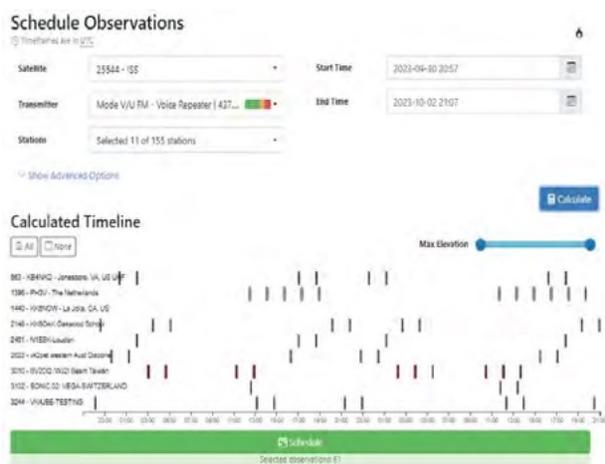
How to Adopt a Satellite:

Where does one start? While this can be done as an individual exercise, it’s better (and less work) as a group exercise, either as a classroom or space club activity.

First explore the SatNOGS.org website. Then assemble a minimal ground station and start scheduling observations on your station. Once your station is on-line, look at the SatNOGS network map and schedule simultaneous observations over several other stations, some nearby and some far away.

Assemble students into teams of 3 to 5. Each team picks a satellite for adoption. Ideal adoptees have a dashboard, are currently active and successfully received by numerous stations, have more than 6 months of on-orbit data available, and documentation is readily available via the owner’s website.

Each team learns as much as they can about their satellite. They schedule downlinks across their global virtual network, monitor received data and compare data to historical data over a 2 to 4 week period. At the end, they present their research and findings to the group. entering and leaving eclipse.



Scheduling downlinks



Typical dashboard

For real excitement, students can adopt a satellite before launch. SatNOGS often publishes information on their forum about ride-share missions pre-launch. Students can often watch SpaceX ride-share deployments in real-time on YouTube or X. CubeSats typically start transmitting soon after deployment. Dozens of SatNOGS station owners may compete for bragging rights to be the first to receive the signal. It can be a fun and involving exercise for students, but frustrating. Expect many observations with no signals.

In Conclusion:

This combined approach offers the **hands-on experience of building and understanding a CubeSat** (via CubeSatSim) and **access to real telemetry with the experience of being a**

satellite operator (via SatNOGS), all without launch risks. This powerful framework fosters deep engagement with space science and engineering. For more information, visit SatNOGS.org, CubeSatSim.org, or Amsat.org.

New CubeSatSim Features and Capabilities

Alan Johnston, KU2Y
AMSAT VP Educational Relations

Introduction

The CubeSat Simulator <https://github.com/alanbjohnston/CubeSatSim/wiki> is a low-cost satellite emulator that runs on solar panels and batteries, transmits UHF radio telemetry, has a 3D printed frame, and can be extended by additional sensors and modules. The CubeSatSim is shown in Figure 1.



Figure 1. AMSAT CubeSatSim Satellite Simulator

Previous AMSAT Journal and AMSAT Space Symposium articles have described the hardware and software. With the recent CubeSatSim v2.1 software release in June 2025, there are some new capabilities that have been added to the CubeSatSim. These new features include: Cross Band FM Repeater, FunCube Mode, and advanced Command and Control (C2C). For full information about this release, see <https://github.com/alanbjohnston/CubeSatSim/releases/tag/v2.1>.

This article will describe the new features and capabilities and also provide instructions on how to try them out with your CubeSatSim. If you don't have a CubeSatSim, you can build one by following the instructions on <https://github.com/alanbjohnston/CubeSatSim/wiki> and purchasing the parts in the BOM <https://CubeSatSim.org/bom> or by purchasing a kit from the AMSAT Store. AMSAT News Service (ANS) will provide announcements when kits will be available.

Updating to the Latest Software

If you have a v1.3.2 or v2 CubeSatSim (blue PCB boards), you can try out these new features and capabilities by updating the software on the Pi Zero inside your CubeSatSim. If your Pi Zero is connected to your WiFi network and hence the internet, you can do it by just logging into your Pi Zero and typing these commands (for information about how to log into your Pi Zero and get it on your WiFi, see <https://cubesatsim.org/download/cubesatsim-readme.pdf>):

```
cd  
cd CubeSatSim  
git checkout master  
./update
```

Note that if your CubeSatSim is an early v1.3.2 version and does not have a USB sound card plugged into the Pi Zero, you will need to make these hardware changes to take advantage of these new features:

<https://github.com/alanbjohnston/CubeSatSim/wiki/Beta-Upgrade-to-v2.0>

The new features are Cross Band Repeater, FunCube Mode, and Enhanced Command and Control, which will be discussed in the following sections.

Note that an amateur radio license is needed to use the Cross Band Repeater mode and Command and Control involves transmitting in the Ham Radio 70cm band. Before doing this, ensure you have your license and comply with all local rules and regulations.

This article also discusses the new Ground Station software Fox-in-a-Box v4 (FIABv4) Beta Raspberry pi disk image.

Cross Band FM Repeater Mode

The CubeSatSim Cross Band FM Repeater mode uses the FM transceiver module in receive mode, with the squelch pin read by the Raspberry Pi Zero to detect a

transmission. The USB sound card connected to the Pi Zero with the OTG cable, and the audio jumper from the Main board to the sound card provides the received audio to the Pi Zero. The Pi Zero then connects the microphone input to the rpitx transmitter. One antenna is receiving the UHF signal, the other is transmitting the VHF signal at the same time, just like a real CubeSat with an FM repeater.

The v2.0 software release had a Repeater mode, but you couldn't turn it on and off with the pushbutton, and it was unstable in that the repeater function would stop working and require a reboot. It also had uplink and downlink frequencies in the UHF band which wasn't ideal.

The v2.1 software release has fixed these issues, and moves the downlink to the VHF band. This provides a "U/V" operation with an FM UHF uplink, VHF downlink similar to many AMSAT CubeSats such as AO-91 (RadFxSat / Fox-1B) and AMSAT-UK AO-73 (FunCube).

To try out the Cross Band Repeater Mode, you will need:

- CubeSatSim with v2.1 software or later.
- Two VHF/UHF FM radios (HTs), or one UHF FM radio and an SDR (such as a Baofeng and an RTL-SDR plugged into a Pi or Computer) or a VHF/UHF transceiver with full duplex (such as an Icom IC-9700). These options are shown in Figure 2.



Figure 2. Radio Options: Two HTs, One HT and one SDR, or One Full Duplex Transceiver

The "full duplex" capability, provided with either two radios, a radio and an SDR, or a full duplex VHF/UHF radio is essential for you to transmit and hear your own signal at the same time.

You can put your CubeSatSim into the Cross Band Repeater mode the same way you set any of the other six modes. You can:

- Use the pushbutton to change to the Repeater Mode 6 by pressing and holding the pushbutton and releasing when the Green Power LED blinks six times.
- Log into your Pi Zero 2 and typing the command **CubeSatSim/config -e** (see <https://cubesatsim.org/download/cubesatsim-readme.pdf> for details of how to log into your Pi Zero 2)
- Use Command and Control (C2C) by sending **6#** with DTMF (Dual Tone Multi-Frequency) or **MODE=e** with APRS (see the section on Advanced Command and Control later in this document).

When you are in the Cross Band Repeater mode, the Green Power LED will be illuminated all the time, but the Blue Transmit LED won't illuminate unless the repeater is active. No telemetry is transmitted after the initial CW ID.

The CubeSatSim will be in receive mode, listening on the default frequency of 435 MHz, or another UHF frequency set by the **CubeSatSim/config -F** command. When the squelch is broken on that frequency, the CubeSatSim will retransmit the FM signal received at the default frequency of 144.9 MHz. Or, if another transmit frequency is set by the **CubeSatSim/config -F** command, the transmit frequency will be the set transmit frequency minus 290 MHz (for example, the default 434.9 MHz – 290 MHz = 144.9 MHz).

The Blue Transmit LED will illuminate as long as the squelch is broken. There is about a one second timeout. If using the Raspberry Pi Ground Station, select the 2m band that covers 145 MHz. On the FIABv4 WebSDR, use the “RTL-SDR 2m Ham Band 145 MHz (CubeSatSim Cross Band Repeater)” band as shown in Figure 3.

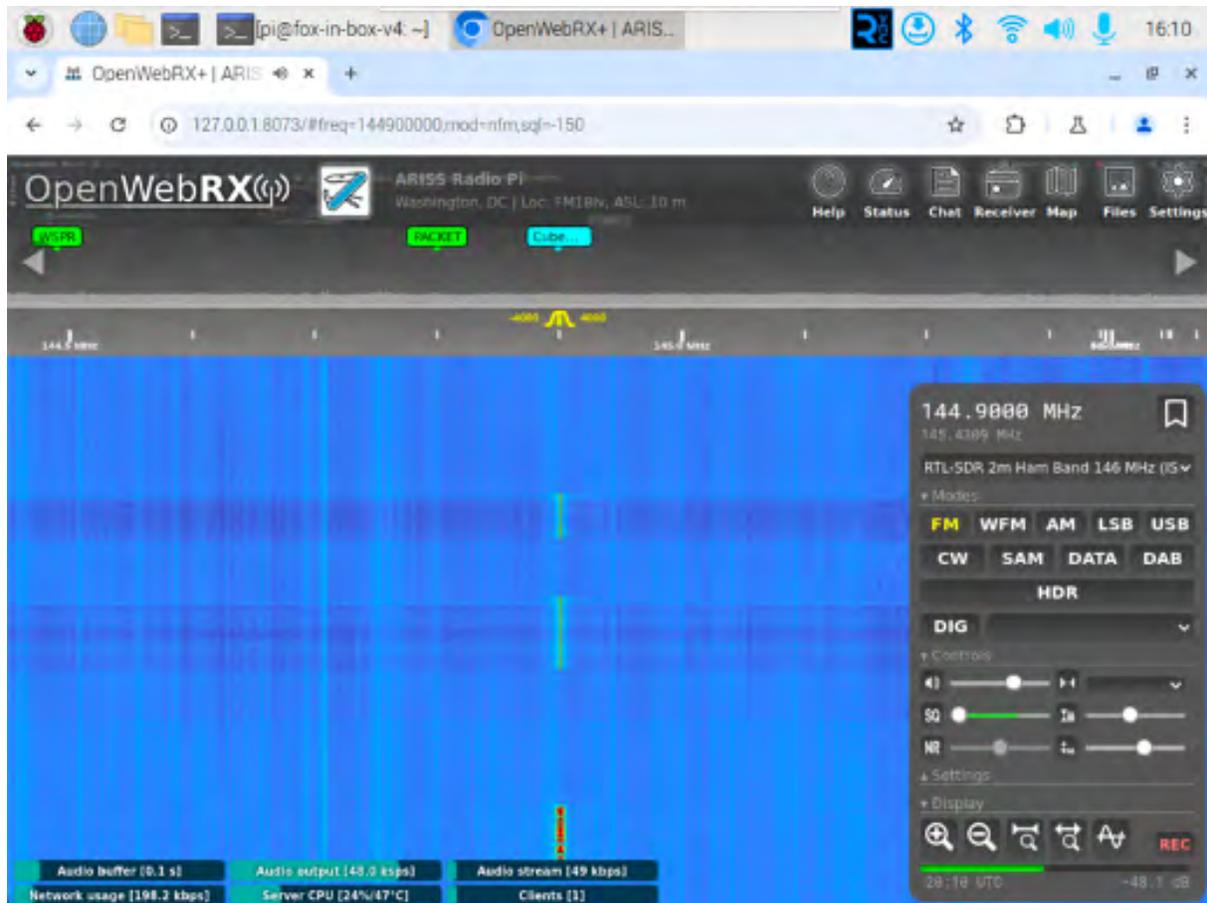


Figure 3. Raspberry Pi Ground Station based on the Fox-in-a-Box image for monitoring the VHF downlink using the OpenWebRX WebSDR.

You should initially listen for the VHF repeated signal with no squelch set on your receiver as the signal will be weak. Also, there will be a small random frequency offset from expected frequency so you may need to tune a few kHz either side of the expected frequency – you may need to reduce your receiver’s frequency step from 10kHz or 15kHz down to 1kHz. An SDR with a waterfall is helpful to find the signal. Each time the CubeSatSim reboots, there will be a different frequency offset, so if there is a big offset, you can try pressing and releasing the pushbutton to reboot the CubeSatSim.

If you need to adjust the squelch setting on the CubeSatSim receiver, you can try different squelch level settings using the **CubeSatSim/config -q** command on the Pi Zero 2.

To prevent false triggers, you can set a CTCSS PL code on the CubeSatSim that will need to be set on the transmitting radio in order to break the squelch. For example, you could set a 67 Hz CTCSS tone just like AO-91 (RadFxSat / Fox-1B). For information see <https://github.com/alanbjohnston/CubeSatSim/wiki/Command-and-Control>

You can change out of this mode the same way you set any mode, such as:

- Use the pushbutton to change to another mode
- Log into your Pi Zero 2 and use the command **CubeSatSim/config** to another mode.
- Use Command and Control (C2C) by sending the mode number then # with DTMF or an APRS packet with **MODE=** and the mode letter (see the Advanced Command and Control section for more information)

If you are trying out the mode and never see the Blue Transmit LED illuminate when you transmit, check to make sure you are transmitting an FM signal at 435 MHz. On your UHF radio or SDR, verify you can hear or see your FM transmission on 435 MHz.

If you see the Blue Transmit LED illuminate, but never hear anything on VHF, check to make sure you are listening on the right receive frequency, 144.9 MHz. The signal can be shifted by a few kHz, so you might need to tune up and down a bit. Make sure your squelch on your receiving radio or SDR isn't set too high - try turning off the squelch and turning the volume up high. The frequency might also be shifted – try going up or down a few kHz from the expected frequency or use an SDR receiver so you can see the carrier to tune accurately.

New FunCube Mode

The CubeSatSim now has a new FunCube Mode 7. This BPSK (Binary Phase Shift Keying) 1200 bps (bits-per-second) transmission mode emulates the AMSAT-UK FunCube CubeSat (AO-73) and several other CubeSats that also use this transmission including Nayif 1 (EO-88), UKube-1, and JY1 Sat (JO-97). The transmission is very similar to the Fox-1E BPSK 1200 bps transmission already supported by the CubeSatSim in BPSK Mode 3. However, the data structure and error correction code needed to be written. Special thanks to Duncan Hills, the lead FunCube Dashboard developer from AMSAT-UK who provided technical details and support to get this code working!

You can see how similar the telemetry is by receiving FunCube Mode in FoxTelem BPSK mode. You will see an open eye and a good phasor diagram for a few seconds, but then it will go to noise, and no frames are decoded. If you listen to it, it sounds similar, except for every four seconds it sounds like there is a “beep” or tone.

To decode FunCube telemetry, you need a FunCube Dongle (FCDPro or FCDPro+) and decoding software (an RTL-SDR dongle will not work). For the FunCube satellites in

orbit, there is a Windows Dashboard for each satellite that decodes the telemetry and uploads the data to the AMSAT-UK Data Warehouse. We don't yet have a Windows Dashboard, but we do have a Raspberry Pi application which will decode it, shown in Figure 4. It is part of the new Fox-in-a-Box-v4 beta image.

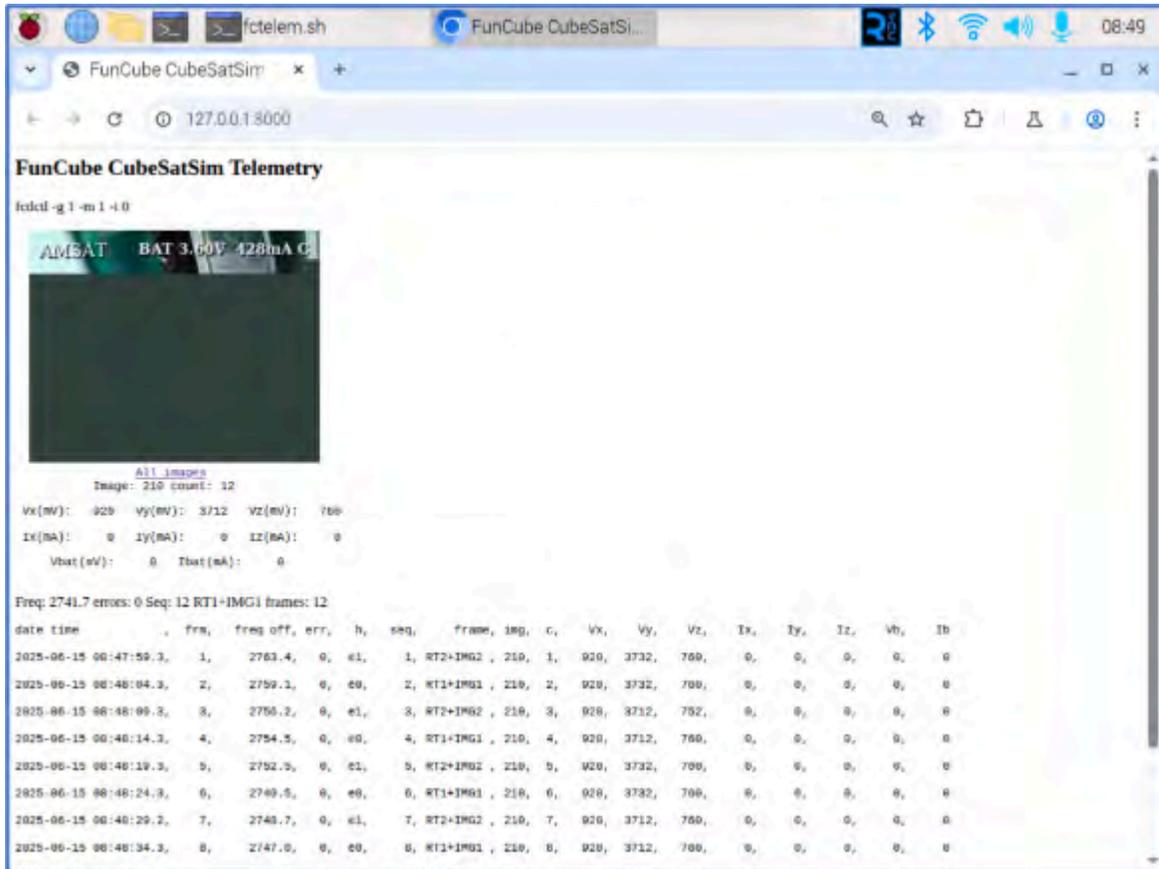


Figure 4. FunCube Telemetry Decoding using a FunCube Dongle on a Raspberry Pi with FIABv4

The telemetry data is a similar format to JY1 Sat (JO-97) with a camera image and telemetry data in each frame.

Unlike the CubeSatSim SSTV mode 4 which uses tones to transmit image pixels in Scottie 2 format, FunCube uses SSDV to encode JPEG images suitable for packetization. It uses the Reed-Solomon codec written by Phil Karn, KA9Q. You can see the image come in block by block, and if a frame is lost, there is a rectangular dropout in the image.

FunCube Mode is mode 7 for the CubeSatSim and can be set using the pushbutton or using the **CubeSatSim/config -j** command.

In the telemetry decoder, the gain of the FCD can be set, along with turning on and off the LNA and Mixer.

Advanced Command and Control

Satellite Command and Control refers to the ability to connect, usually over radio, with a satellite and upload commands, settings, schedule operations and experiments, and other functions. The CubeSatSim v2 can now support radio Command and Control using a FM transmitter in the UHF band by sending either a carrier signal, DTMF tones or APRS packets to change modes or turn the transmitter on or off.

To use Command and Control, you will need a v1.3.2 or v2.0 CubeSatSim with the USB soundcard and an UHF FM radio or transmitter.

The default frequency which you should transmit on for Command and Control is 435 MHz FM. You can change this frequency by logging into your Pi Zero 2 and typing **CubeSatSim/config -F** and changing the UHF receive frequency of the CubeSatSim.

By default, Command and Control is OFF on the CubeSatSim. There are two ways to turn it on or off:

- Using the pushbutton. Press and hold the pushbutton until it goes through its full sequence: a series of short blinks, some longer blinks, then some even longer blinks. At this point, you can release the pushbutton. This will toggle the state. That is, if Command and Control was ON, it will turn off. If it was OFF, it will turn ON.
- Using a command line on your Pi Zero 2. You will need to log into your Pi Zero 2 (see <https://cubesatsim.org/download/cubesatsim-readme.pdf> for how to do this) then type the command **CubeSatSim/config -T** and then follow the prompts to turn it on or off.

You can tell if Command and Control is ON if the letter “C” is transmitted in CW after the callsign on power up.

When turned on, Command and Control has three modes:

- *Carrier (Squelch) Command and Control*. In this mode, the CubeSatSim is just listening for a carrier signal on the receive frequency. The squelch setting on the FM Transceiver set with the **CubeSatSim/config -q** will determine the sensitivity. If a carrier is detected, the CubeSatSim will move to the next mode and the Green Power LED will blink the number of times of that mode. The sequence is APRS -> FSK -> BPSK -> SSTV -> CW

-> Repeater -> FunCube -> APRS and the cycle repeats. This mode is the simplest, but it can produce false triggers.

- *DTMF Command and Control.* In this mode, the CubeSatSim is listening for DTMF key presses followed by the pound (hash) sign #. The DTMF number corresponds to the mode number. The Green Power LED will blink with the number of the mode and then the CubeSatSim will restart or reboot. The set of commands is listed in the table.
- *APRS Command and Control.* In this mode, the CubeSatSim is waiting for an APRS packet containing the string **MODE=** followed by a letter of the new mode. The Green Power LED will blink with the number of the mode and then restart. The set of commands is listed below:Figure 5.

DTMF Command	APRS Command	Result
1#	MODE=a	Change to APRS Mode (mode 1)
2#	MODE=f	Change to FSK Mode (mode 2)
3#	MODE=b	Change to BPSK Mode (mode 3)
4#	MODE=s	Change to SSTV Mode (mode 4)
5#	MODE=m	Change to CW Mode (mode 5)
6#	MODE=e	Change to Repeater Mode (mode 6)
7#	MODE=j	Change to FunCube Mode (mode 7)
10#	MODE=o	Turn transmission of telemetry on or off



Figure 5. Command and Control Commands.

When Command and Control is turned on the first time, the default mode is DTMF or APRS mode. To change it to Carrier (Squelch), log into your Raspberry Pi and enter the command **CubeSatSim/config -d**

To troubleshoot Command and Control, you can look at the Command and Control log file on the Raspberry Pi Zero 2 by typing: **CubeSatSim/log -c**

You can change the frequency that the CubeSatSim listens on to another frequency in the UHF band by logging into your Pi Zero 2 and typing the command **CubeSatSim/log -F**

You can change the squelch level that the CubeSatSim receiver uses by logging into your Pi Zero 2 and typing the command **CubeSatSim/log -q**

The default Command and Control has no security built in. Anyone can transmit on the frequency and the CubeSatSim will respond. You can add a small amount of security by activating CTCSS PL (Private Line) codes on the FM receiver by logging into your Pi and typing **CubeSatSim/config -P** You will then need to program the same PL code into your FM transmitter.

When you power up the CubeSatSim, if you hear a Morse Code “C” after the callsign, this means Command and Control is enabled. If you are in APRS mode the letter “C” after the telemetry string in the packet indicates that C2C is enabled. In SSTV mode, the letter C after the callsign on camera images indicates that C2C is enabled.

Fox-in-a-Box v4 Beta Ground Station Software

The Fox-in-a-Box (FIAB) Raspberry Pi disk image has been available from AMSAT for a long time. It has always provided a Raspberry Pi disk image with FoxTelem pre-installed for receiving telemetry from AMSAT’s CubeSats, starting with Fox-1A (AO-85) through MESAT1 (AO-122). FIAB runs on a Raspberry Pi with a FunCube or RTL-SDR dongle and an antenna, and can be used to receive, decode, and upload telemetry.

Starting with the FIABv3, the ARISS Radio Pi project was merged with FIAB which not only provided FoxTelem but other satellite and SDR software including:

- Direwolf for APRS decoding
- QSSTV for SSTV decoding
- CubicSDR for general SDR operation
- OpenWebRX as a WebSDR, a minimal configuration web-based SDR interface
- Gpredict for tracking satellites
- RTL-TCP for connecting to an RTL-SDR dongle over the internet
- KLAtracker for monitoring satellite passes
- PacSat a ground station for Packet Satellites

The new FIABv4 Beta image, shown in Figure 6, has all the capabilities of FIABv3 in addition to:

- Support for the Raspberry Pi 5
- Support for the RTL-SDR v4 (except in FoxTelem)
- OpenWebRX+ which adds built in decoders for SSTV and CW modes
- FunCube Telem, a new experimental telemetry decoder for FunCube mode
- SDR++ an open source SDR which has the same user interface in Windows and Mac as on the Pi

As before, the FIABv4 image is available for download so that you can write it to a 16GB or larger micro SD card. <https://cubesatsim.org/download/fiabv4-beta-5.iso.zip> It is also available for purchase on the AMSAT Store.

The FIABv4 Raspberry Pi Ground Station software is recommended for the CubeSatSim as it has the FunCube Mode 7 telemetry decoder.



Figure 6. Fox-in-a-Box v4 Beta Raspberry Pi Image

Building AMSAT CubeSatSims in the Classroom: Experiences with Freshman Engineering Students

ALAN JOHNSTON, PhD, KU2Y
Vice President, Educational Relations, AMSAT
Associate Teaching Professor, Villanova University
ku2y@arrl.net

EDWARD CHAR
Professor of the Practice, Villanova University
edward.char@villanova.edu

Abstract- The AMSAT CubeSatSim is a small, low cost, spacecraft simulator that serves as a tool for academic education, public demonstrations and theory & design disciplines. It can be used in a classroom, lab or training setting to introduce the basics of satellite operations, or it can be used to teach STEM (Science, Technology, Engineering, Math) exercises. This article is about a seven week mini project at Villanova University in spring of 2022 in which freshman electrical and computer engineering students built six CubeSatSims in three classes. The class design, organization, and the outcomes are discussed in this article.

CubeSatSim Overview

A CubeSat is a nano satellite designed for low cost educational space missions, defined by the CubeSat Design Specification (CDS) <https://www.cubesat.org>. The AMSAT CubeSatSim, the CubeSat Simulator, shown in Figure 1, is a Raspberry Pi Zero-based, 3D printed, functional model of a 1U form factor CubeSat. It is designed to act, as reasonably as possible, as 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 <https://cubesatsim.org>. Like real LEO satellites, this simulator is self-powered through onboard rechargeable batteries and solar panels. It transmits telemetry wireless on the ham radio UHF band in a variety of formats. For details on the design and construction of the simulator, see our article in the March/April 2020 issue of the AMSAT Journal

(<http://cubesatsim.org/content/CubeSatSimPaper5.pdf>). There is also a lot of good information available at the CubeSatSim Project page <https://cubesatsim.org>.

For a history of the AMSAT CubeSat Simulator, see the article “A Year with the AMSAT CubeSat Simulator: 12 Months in the Classroom and Lab” in the AMSAT 2019 Space Symposium.



Figure 1. The AMSAT CubeSatSim with low cost 5.5V solar panels, 3D printed frame, and tape measure dipole antenna.

A high level block diagram of the CubeSatSim design is shown in Figure 2.

CubeSatSim Block Diagram v1

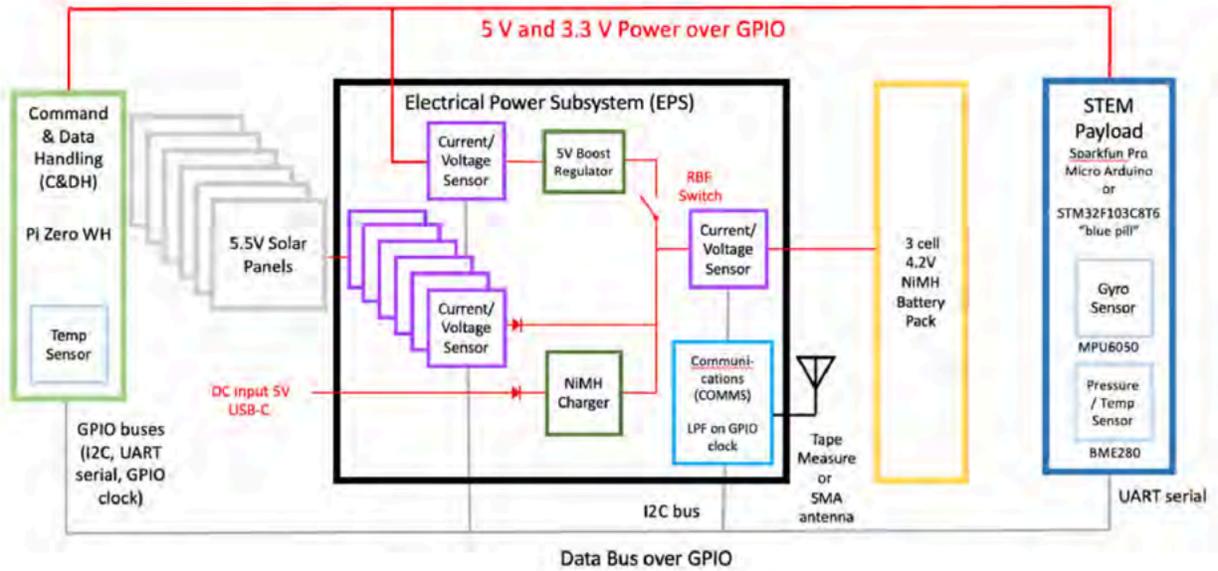


Figure 2. Block Diagram of the CubeSatSim

The four boards of the CubeSatSim are shown in Figure 3.

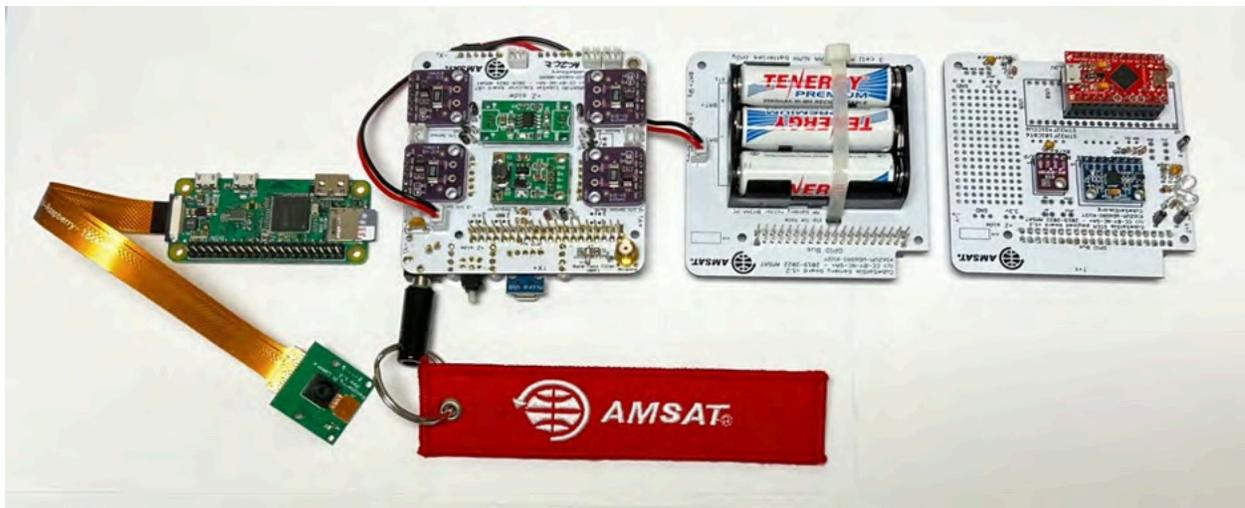


Figure 3. Boards of the CubeSatSim. From left to right, the Pi Zero WH with Pi Camera, Main board, Battery board, and STEM Payload board with Sparkfun Pro Micro microcontroller.

A Ground Station is used to receive the radio telemetry and decode and display the information, such as solar panel and battery voltages, currents, temperatures, and camera images, and sensor and gyro data. A low-cost (\$35) RTL-SDR Software Defined Radio (SDR) USB dongle is used along with open source software. The software is the same software the AMSAT and

other satellite builders use to track and monitor on-orbit spacecraft. The software can run on a PC or on a Raspberry Pi 3B or 4B.

This article is about how CubeSatSims were built in a classroom setting by students organized into subteams that built and tested specific parts of the simulator.

Course Overview

The course was an introductory electrical and computer engineering course taken by students in their second semester of their four year degree program. The goal of the course is to give students some hands-on experience doing engineering work in a project setting.

Here are the specific goals of the mini project as presented to the students:

- Learn about CubeSats and satellite technology
- Understand the role of radio and telemetry in satellite operation, and how to interpret telemetry
- Gain experience working in small teams on different aspects of a project
- Get experience giving a presentation, writing a short report, and sharing information using a Wiki
- Have some fun!

Earlier in the course, students had completed a two week soldering project in which they built and tested a flashing and buzzing PCB board. Nearly all students had never soldered before, so the CubeSatSim was their second soldering project.

The CubeSatSims were built as part of a CubeSat Mini Project which also included the following activities:

- Researching CubeSats and launch vehicles.
- Building tape measure Yagi-Uda antennas and listening to the ISS and satellites on passes over campus using laptops and RTL-SDR dongles.
- Giving a short impromptu “Space Hot Topics” presentation.
- In previous semesters, we also did a SatNOGS activity where students selected satellite passes for observations in one class then listened to the results and vetted them in another class. This activity is described in Appendix A.

Subteams

There were three classes in total, and the class size ranged from 18 to 30 students. Each class was divided into two groups, and each group built a CubeSatSim. Students in each group were

divided into subteams who had responsibility for a particular subsystem, as shown in Figure 4, which was a slide presented to students.

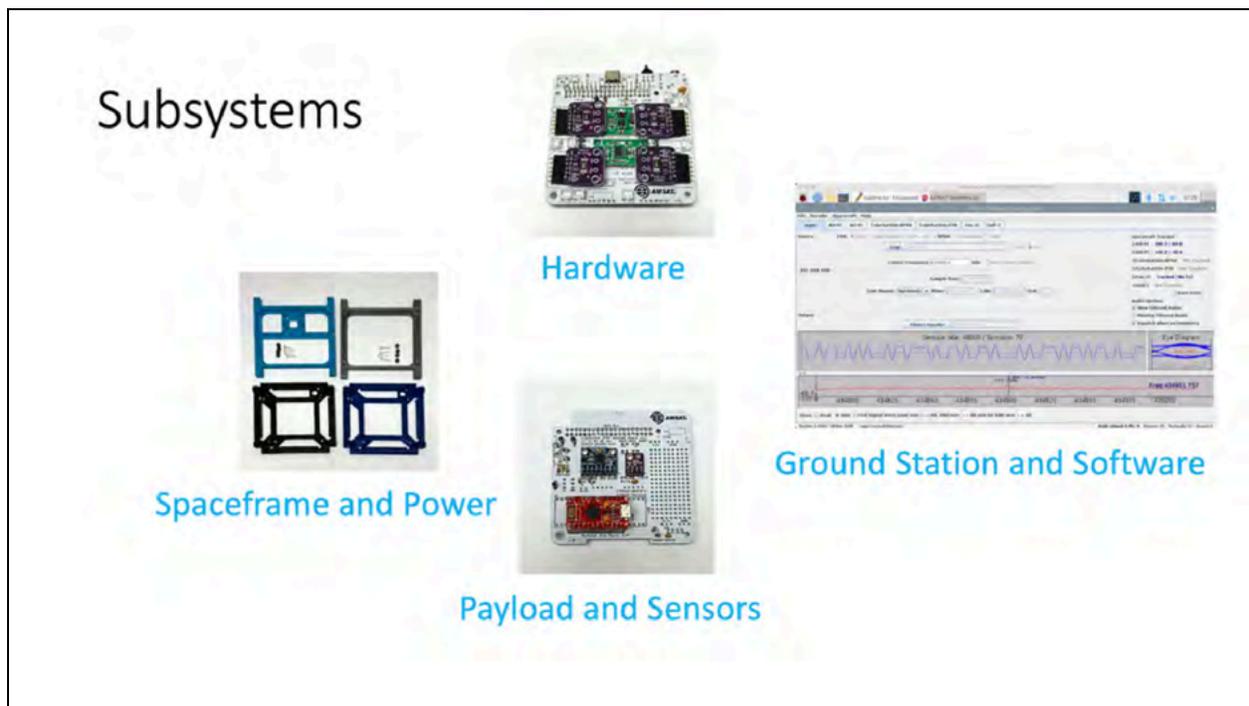


Figure 4. Four Subsystems of the CubeSatSim

Here are the details of the subteams and their responsibilities:

- *Spaceframe and Power* - 3D printing the frame, soldering and testing the solar panels, building and mounting the antenna
- *Hardware* - soldering and testing the Main board
- *Payload Sensor* - soldering and testing the STEM Payload Board, using the Arduino IDE to install the software for the Pro Micro microcontroller
- *Ground Station and Software* - installing the software for the Raspberry Pi Zero, configuring it for testing, installing and using ground station software on PCs and Raspberry Pis

Students had a chance to indicate their preferred subteam. Approximately half of the students expressed a preference, so everyone's first preference was satisfied.

As the size of classes varied, and we were building two CubeSatSims per class, the size of the subteams varied from 2 to 4 students per team.

Steps in Building a CubeSatSim

As listed in the Wiki build instructions <https://CubeSatSim/wiki>, there are the nine steps to building a CubeSatSim.

For these nine steps, the primary subteam responsible for each step is listed after each one:

1. [Build the Main board Part 1](#) (Hardware)
2. [Install the Software](#) (Ground Station and Software)
3. [Build a Ground Station](#) (Ground Station)
4. [Continue Building the Main board Part 2](#) (Hardware)
5. [Build the Battery board](#) (Spaceframe and Power)
6. [Assemble the Solar Panels and Frame](#) (Spaceframe and Power/Ground Station and Software)
7. [Finish the Main board Part 3](#) (Hardware)
8. [Build the STEM Payload board](#) (Payload and Sensor)
9. [Put the Board Stack together and mount in the Frame](#) (All)

This is also the recommended order if a single person or small group is building a CubeSatSim. However, when the build is being done across subteams, the work breakdown schedule is shown in Table 1, based on a 75 minute class build period.

Build Session	Hardware Subsystem	Spaceframe and Power Subsystem	Ground Station and Software Subsystem	Payload and Sensors Subsystem
1	1. Main Board 1 Build	6.2 3D printing frame	2. Software Install	Arduino IDE install and Blink Test (Blink test part of 8.2 STEM Payload Board)
2	1. Main Board 1 Build	5. Battery Board Build	3.2 PC Ground Station	8.1 STEM Payload Board Build
3	4.1 Main Board 2 Testing 4.2 Main Board 2 Build	6.1 Solar panels	3.1 Raspberry Pi Ground Station	8.1 STEM Payload Board Build
4	4.2 Main Board 2 Build	6.1 Solar panels	6.3 Pi Camera Install	8.2 STEM Payload Board Testing
5	7.1 Testing Main Board 3 7.2 Main Board 3 Build	9.1 Antenna Install	Simulated Telemetry Analysis	Sensor Verification on PC and on Telemetry
6	7.3 Flat Sat Testing 9.2 Stacking Boards	7.3 Flat Sat Testing 9.2 Stacking Boards	7.3 Flat Sat Testing 9.2 Stacking Boards	7.3 Flat Sat Testing 9.2 Stacking Boards
7	9.3 Install in Frame	9.3 Install in Frame	9.3 Install in Frame	9.3 Install in Frame

Table 1. Build Schedule by Subteam.

The numbers for each item in the Table refer to the step number in the CubeSatSim Wiki instructions. For example, “4.2 Main Board 2 Build” is Step 4 section 2 in the Wiki which is called “Main Board 2”.

Some steps have no dependencies; for example the Main board, Battery Board, and STEM Payload board can all be built independently, as well as the Pi Zero software install. However, there are dependencies in testing which this breakdown tries to address such as:

1. The testing of the Main board in Step 4.1 requires the Pi Zero with the software installed.
2. The testing of the INA219 boards in Step 4.2 requires the Pi Zero with the software installed.
3. The testing of the Main board in Step 7.1 requires a Ground Station, Battery board, Pi Zero with software installed, and a solar panel and connector.
4. The testing of the STEM Payload board in Step 7.2 Complete Payload Test requires a Pi Zero and a Ground Station

5. The Flat Sat testing in Step 7.3 requires a Ground Station, Battery board, Pi Zero with software installed, and a solar panel and connector.
6. The Stacking Boards step 9.2 requires all three boards and the Pi Zero.
7. The Installation of the Boards in the frame step 9.3 requires all boards and the frame.

Each subteam had a Wiki page which they used to record their progress, identify issues and problems. In addition, the instructors used the Wiki pages to communicate with the subteams. Before each class, instructors posted guidance and other information to students who were advised to read their Wiki pages at the start of class. With 15 minute remaining in each class, students were reminded to fill in their Wiki pages with their status updates.

Each classroom also had:

- A complete CubeSatSim for reference. The Ground Station and Software teams used it during the initial weeks to generate radio signals and learn how to operate the CubeSatSim
- A CubeSatSim Lite board with a stacking GPIO header installed. It was plugged into the Pi Zero WH by the Ground Station and Software teams to test radio transmission during the early sessions. The STEM Payload board was plugged on top to test the STEM Payload Board as well.
- A complete Raspberry Pi Ground Station with a touch screen with speakers and wireless mouse. This provided decoding of all telemetry modes.

Here is a list of the tools which were provided in each classroom:

- Soldering iron and lead free solder (to solder the components on the boards)
- Metal curls solder tip cleaning station (to clean the soldering iron tip)
- Safety glasses (to protect eyes while soldering or trimming leads)
- Small Phillips screwdriver (for standoff hardware)
- Small flathead screwdriver (for frame hardware)
- Needle nose pliers (to bend leads and hold parts)
- Heat gun (for heat shrink tubing)
- Side cutters (to trim leads)
- Drill and drill bits, sandpaper or emery cloth (for making the dipole or monopole antenna and Remove Before Flight plug)
- Liquid flux (to solder SMA connector)
- Various micro USB and USB-C cables
- Micro SD card USB writer to flash the micro SD card for the Pi Zero W

Kits

The online CubeSatSim BOM (Bill of Materials) has a sheet "By Step" (click at the bottom - see Figure 5) which lists the parts needed for each step in order. <http://cubesatsim.org/bom> If

you have a Google account, you can make a copy of this spreadsheet ("File" then "Make a Copy") and check off each part as you install it.

Bill of Materials for AMSAT CubeSatSim v1.2

File Edit View Insert Format Data Tools Extensions Help Last edit was on June 28

100% \$ % .0 .00 123 Calibri 14 B I A

A1

1 **Bill of Materials for CubeSatSim v1.2 for info see** <http://cubesatsim.org>

2

3 Note: if you have the v1.0 PCBs, here is the BOM: <https://docs.google.com/spreadsheets/d/1UQdsXe-uZ38PD8IU8TVegTDz>

4 Note: if you have the v1.1 PCBs, here is the BOM: <https://docs.google.com/spreadsheets/d/1v94Ut4WInf17ZC4sU9S576i>

5

6 *If you have a Google account, you can make your own copy of this spreadsheet by going under File then Make a Copy*

7 *The colored rows which begin with "Alt" indicate alternatives. Uncheck the a choice to get the b choice*

8

9 Main Board <https://github.com/alanjohnston/CubeSatSim/blob/master/hardware/v1.1/cubesim>

10

Ref	Item	Qty	Cost	Total	Source	Source URL	Alternate	
<input checked="" type="checkbox"/>	U10	5V boost regulator 500 mA	1	\$4.00	\$4.00	eBay	https://www.ebay.com/sch/l	https://www
<input type="checkbox"/>	U9	NiMH CN3085 3s charger	1	\$2.99	\$2.99	eBay	https://www.amazon.com/Vil	https://www
<input type="checkbox"/>		1x14 male breakaway header	1	\$0.47	\$0.47	Digikey	https://www.digikey.com/en/r	
<input type="checkbox"/>		USB-C charger with Micro USB adapter	1	\$10.99	\$10.99	Amazon	https://www.amazon.com/	
<input type="checkbox"/>	X1	SC1464-ND 3.50mm Headphone Phone Jack S	1	\$2.10	\$2.10	Digikey	https://www.digikey.com/g	https://www
<input type="checkbox"/>	D1-D7	Diode 1N5817RLG D1, D2, D3, D4, D5, D6, D7	7	\$0.40	\$2.80	Digikey	https://www.digikey.com/g	https://es.e
<input type="checkbox"/>		Remove Before Flight Tag	1	\$8.00	\$8.00	Amazon	https://www.amazon.com/	https://www

By Board By Step

Figure 5. CubeSatSim online Bill of Materials spreadsheet <https://CubeSatSim.org/bom>

We started with CubeSatSim PCB Sets from the AMSAT Store <https://www.amsat.org/product/amsat-cubesatsim-pcb/> as shown in Figure 6, which has the Low Pass Filter and USB-C surface mount components already installed, so only through hole soldering is required.

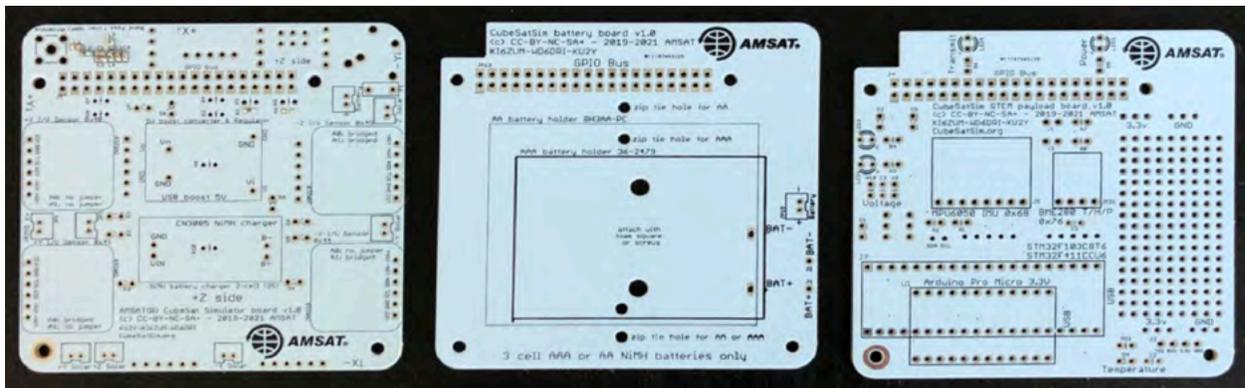


Figure 6. Three blank PCB boards with SMD (Surface Mount Devices) already installed, available at the AMSAT Store.

We ordered Pi Zero W kits which included a blank 64 GB SD Card (a 16 GB is the minimum necessary), power supply, HDMI and USB adapters, and GPIO pin headers. The instructor soldered the pin headers onto the Pi and tested them. The Pi Zero WHs should be reusable for future classes and projects.

We ordered Sparkfun Pro Micro 3.3V microcontrollers, and the instructor soldered the pin headers. The Pro Micros should be reusable for future classes and projects.

We used brass standoffs as they are stronger than nylon ones.

Students used PLA 3D printers available on campus to print the frame after downloading the STL files from the internet. No CAD work was required.

In the BOM spreadsheet, there is a “By Step” view (sheet) that lists the parts needed for each step and also has a photo of each part to aid identification. Parts for each step were put in an anti-static bag and labeled with the Step number. The anti-static bags (approximately 9” x 8”) were large enough for the built parts to also be stored in them. A box for each group contained the nine bags for a complete CubeSatSim. For example, the parts list for Step 7 is shown in Figure 7.

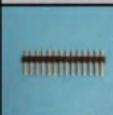
<input type="checkbox"/>		Step 7. Main Board 3			https://github.com/alanbjor
<input checked="" type="checkbox"/>	Ref	Item	Qty	Location	Image
<input type="checkbox"/>		1x14 male breakaway header (1x8 piece)	1	Top	
<input type="checkbox"/>	U9	NiMH CN3085 3s charger	1	Top	
<input type="checkbox"/>		cut off leads from the diodes D1 - D7	4	Top	
<input type="checkbox"/>	U10	5V boost regulator 500 mA	1	Top	
<input type="checkbox"/>	D8	5.1V 400mW Zener	1	Top	
<input type="checkbox"/>	F1	PTC resettable fuse, 1.2A	1	Top	
<input type="checkbox"/>		Remove Before Flight Tag	1		
<input type="checkbox"/>	X1	1/8" 3.5mm Plug Audio Jack Connector	1		

Figure 7. Parts list for Step 7 Main Board 3 in the BOM in By Steps view.

Note that parts that install on the PCB are labeled as Top or Bottom depending on which side of the PCB they are installed on. The PCB also has solder mask on the part side of most components, making it harder to install and solder the components on the wrong side.

An example parts kit for Step 7 is shown in Figure 8.



Figure 8. Parts kit for Step 7.

In each class, one kit contained the tape measure antenna and one kit contained the SMA antenna.

Results

The results of the class CubeSatSim build were encouraging. All but one group completed their tasks and had working boards. Photos of students building and testing the CubeSatSims are in Figures 9 - 12.

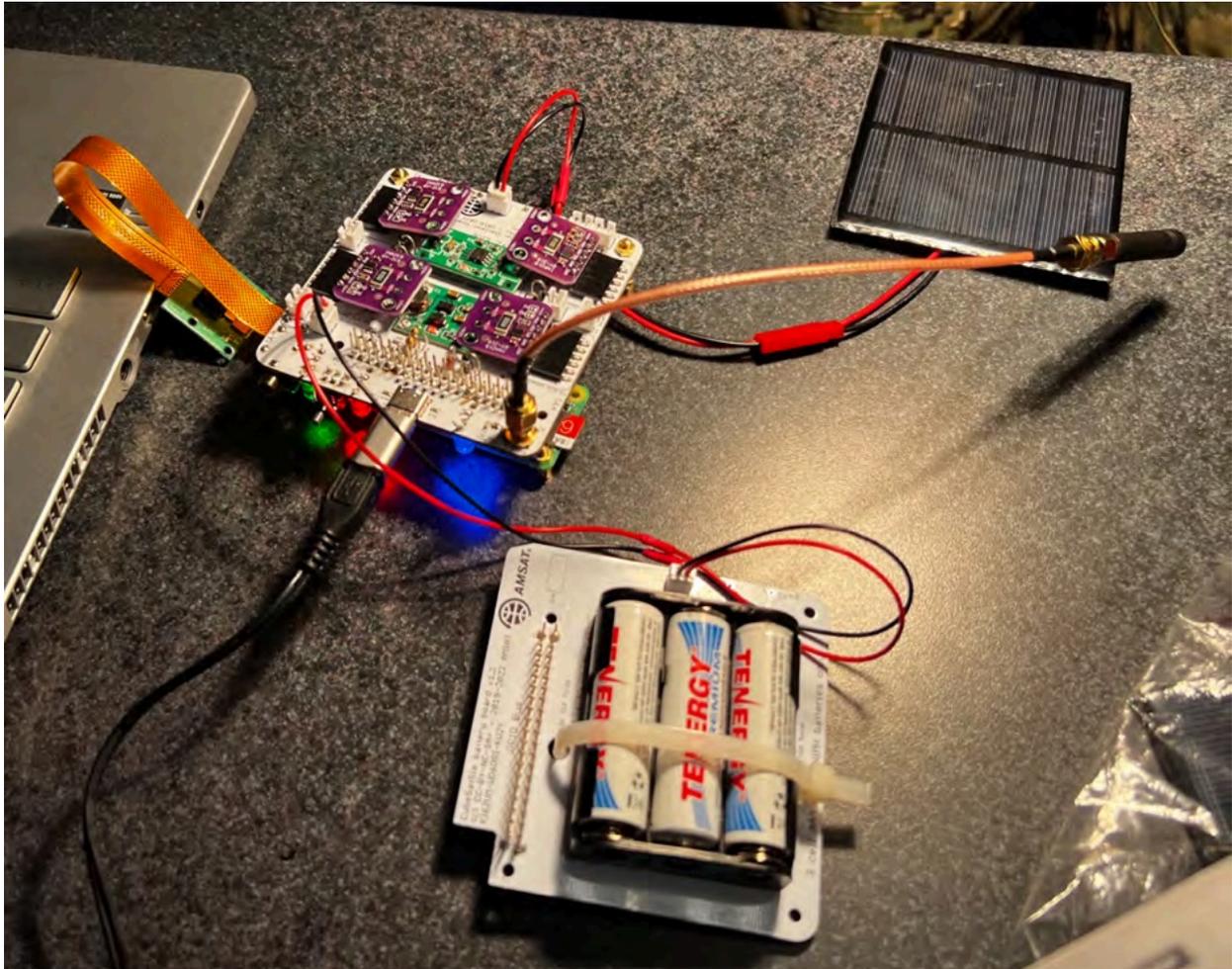


Figure 9. Flat Sat testing of the CubeSatSim.

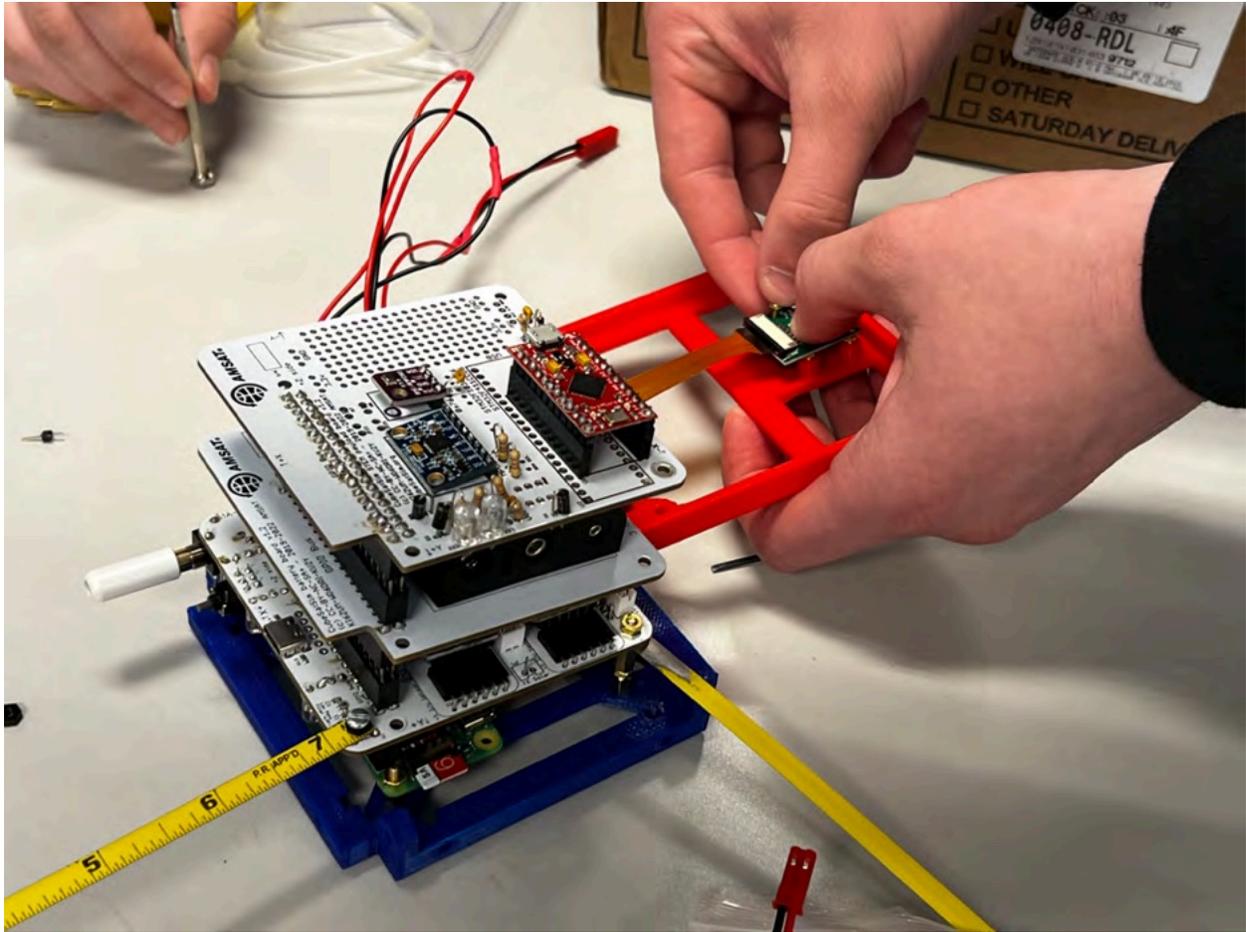


Figure 10. Mounting the camera in the frame.

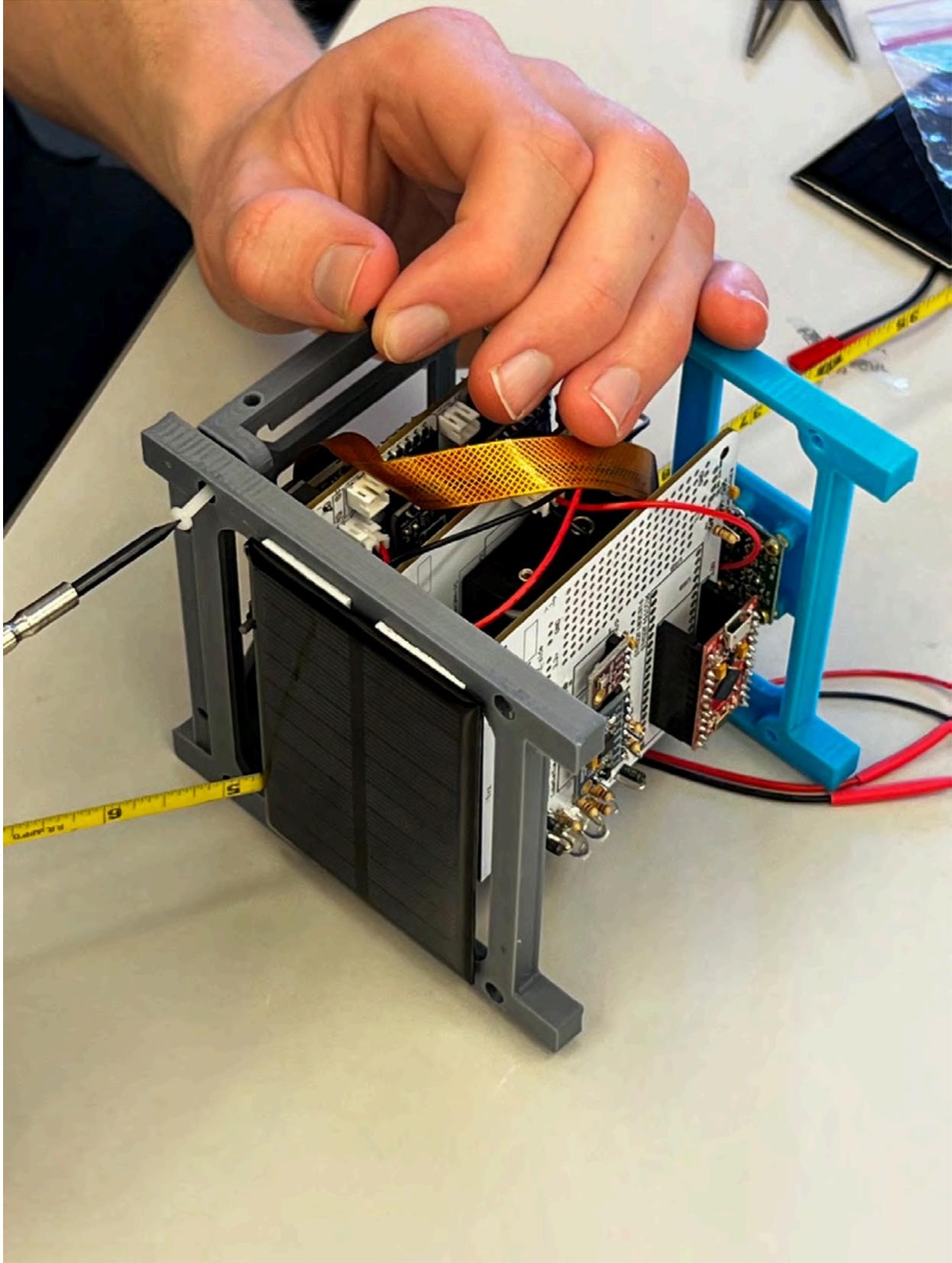


Figure 11. Putting the frame together.



Figure 12. Cart with four completed CubeSatSims

Using the CubeSatSim

The mini project was done at the end of the semester, so there was no opportunity for the schedule to slip. As a result, we unfortunately had little time to use the built CubeSatSims. However, we did have one session where the students received telemetry and interpreted it.

One working CubeSatSim was placed on a turntable in front of the lamp and put on a cart. After “launching” it in front of the class, it was wheeled into the hallway where it couldn’t be seen.

Students ran FoxTelem on their laptops and received radio signals and interpreted the results.

The details of this class activity are described in Appendix B.

Lessons Learned

As this was the first time building the CubeSatSim with the v1 hardware, we learned quite a bit during the project. We realized we needed more time for FlatSat testing and Stacking steps. We did complete the testing and stacking but it was very rushed, and we left out mounting hardware and other non-essential parts.

A few teams could have benefited from extra “catchup” sessions to help them finish their board. The Payload and Sensors teams finished early, so this subteam needs some additional tasks.

Teams of 4 weren’t ideal, resulting in downtime for some team members. Teams of 2 seemed ideal, with both team members fully engaged. Instead of having teams of 4, an additional CubeSatSim should be built in the class to keep the team sizes as small as possible.

Students wanted to edit Wiki pages at the same time, but only one could at a time. One possible solution is to break Wiki pages into parts so they could edit each part individually without conflict.

CubeSatSims powered up in the same classroom interfered with each other and sometimes students were confused by this. A script could be written to change the transmit frequency of one. We also were able to pick up weak signals from CubeSatSims in another classroom that was occurring at the same time.

We had many problems with the purple INA219 boards and testing them was tedious. We now recommend the blue INA219 boards which are easier to find and do not have all the failures.

Students wanted to learn more about other subteams. In the future, we will add a week of cross-training, where one subteam will train another subteam on their subsystem.

One team did soldering on a Battery Board with the batteries installed - they had already installed the zip tie and didn't want to cut it and replace it. During this soldering, they accidentally bridged the + and - pins, and the PCB trace glowed red like a toaster element. The instructor saw (and smelled!) it and quickly unsoldered the bridge. The batteries and PCB still worked. This is a good reminder why we use NiMH batteries instead of Lithium batteries!

The BME sensor was installed upside down on two Payload boards. Now, the PCB has a circle on the silkscreen to help prevent this.

"Unbridging" pins with a soldering iron is an essential skill which many of them developed. Students were reminded to have an instructor inspect a board before doing any testing or applying power.

One group accidentally switched the blue and green LEDs. We left them that way.

One group had trouble at the start with putting the wrong resistors and the wrong polarity of the LEDs. Eventually, they were given a new PCB and just started again. Always have extra parts and PCBs in case this happens.

One team did a perfect job on their Payload board until the last step when they soldered the GPIO connector on the wrong side of the PCB. Unsoldering it would have been difficult (all 40 pins!), so the instructor tacked another GPIO connector on the correct side and had the students solder the remaining pins. This could actually be useful for prototyping, as it would allow jumper wires to be connected.

Conclusion

The CubeSat Mini Project for freshman electrical and computer engineering students was a success. Students worked in teams and built a subsystem then participated in integration testing and system testing. While not every group completed their CubeSat build, all the groups gained some hands-on experience in a realistic engineering project. The six CubeSatSims that were built will be used for demonstrations, capstone projects, and as AMSAT loaners.

Participating in the CubeSatSim Project

You can participate in the AMSAT CubeSatSim Project! Here are some of the ways:

- Document your CubeSatSimulator build and testing on social media

- Offer to demonstrate your CubeSat Simulator to local schools, your ham club, Makerspace or Hackerspace gatherings, or other STEM events

The official CubeSatSim Project Twitter account is [@CubeSatSim](https://twitter.com/CubeSatSim) and we use the hashtag [#CubeSatSim](https://twitter.com/CubeSatSim) to highlight our activities.

Acknowledgements

Thanks to the Villanova University Electrical and Computer Engineering Department who funded the parts and tools for the project.

Thanks to the members and board of directors of AMSAT for their financial support of the CubeSatSim project. Also thanks to the many volunteers and team members who have contributed, especially Jim McLaughlin, KI6ZUM, David White, WD6DRI, Paul Graveline, K1YUB, Fredric Raab, KK6NOW, Mark Samis, KD2XS, Sopwith, N1SPW, Lindsay White, KI6LZN, Randy Standke, KQ6RS, and Pat Kilroy, N8PK.

This project would not be possible without many open source hardware and software projects.

Appendix A - CubeSat SatNOGS activity

Satellite Observation Scheduling Activity

General Directions: You will work individually on this activity. In the first class period, you will choose a satellite which will be passing over Villanova in the next 24 hours. You will research the expected radio signal from that satellite. The satellite observation will be scheduled on the Villanova University CubeSat Club SatNOGS (Satellite Networked Open Ground Station). In the second class period, you will view the observation of your satellite and determine if the satellite was transmitting a signal over Villanova.

We will be scheduling observations of satellites on the Villanova University SatNOGS station, which is located on the roof of Tolentine Hall. Here is the link to access the station:

<https://network.satnogs.org/stations/1159/>

SatNOGS NETWORK Home About Observations Ground Stations Community Wiki 19:03 UTC Sign Up / Log In

1159 - Villanova CubeSat Club W3YP

Owner	w3yp
QTH Locator	FN20ha
Coordinates	40.038°, -75.349°
Altitude	12 m
Min Horizon	10°
Antennas	VHF Lindenblad
Success Rate	1495
Observations	4 months, 1 week ago
Creation Date	1.0
Client version	Last seen 0 minutes ago
Online	95 days, 21:44:14
Uptime	



Built by Villanova University's CubeSat Club. Utilizes a Lindenblad antenna mounted on roof of Tolentine Hall.

Calculate Future Passes

© 2014-2020 Libre Space Foundation.
Observation data are freely distributed under the CC BY-SA license.

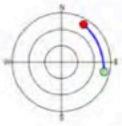
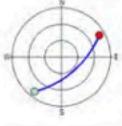
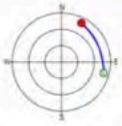
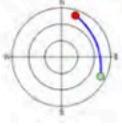
SatNOGS | Back to top
Version: 1.39

Note that all times on SatNOGS are in UTC (which stands for Coordinated Universal Time (why?), the time formerly known as Greenwich Mean Time or GMT, also known as Zulu time). You can see the current UTC in the top right of the website, circled in the above image.

1. Choose a Satellite Pass (10 minutes)

To see passes of satellites over Villanova that can be received by this satellite ground station, click on the “Calculate Future Passes” button at the bottom left of the window (see above).

The website will run calculations based on the current satellites tracked by SatNOGS and the ground station location and display a long listing of satellites passing over in the next 24 hours. Note that your screen will show different satellites depending on when you click on the button:

Name	AOS	LOS	$\uparrow \rightarrow \downarrow$	Polar plot
40908-LILACSAT 2 	21:01 2020-03-27	21:05 2020-03-27	$\uparrow 103^\circ$ $\rightarrow 15^\circ$ $\downarrow 30^\circ$	
25544-ISS 	21:03 2020-03-27	21:10 2020-03-27	$\uparrow 218^\circ$ $\rightarrow 56^\circ$ $\downarrow 60^\circ$	
39444-FUNCUBE-1 	21:09 2020-03-27	21:14 2020-03-27	$\uparrow 104^\circ$ $\rightarrow 16^\circ$ $\downarrow 27^\circ$	
39439-FIRST-MOVE 	21:12 2020-03-27	21:19 2020-03-27	$\uparrow 116^\circ$ $\rightarrow 21^\circ$ $\downarrow 19^\circ$	

The key information in each row is circled above and is:

- Name: The NORAD ID and name of the satellite. In this example, 40908 and Lilacsat 2
- Success Rate: A graphical indication of the ratio of Successful Observations (Green) and Bad Observations (Red). Sometimes you might also see Unvetted Observations (Yellow) which haven't been classified yet. If you mouse over the green part, you get the Success Rate in percentage, in this case a low 27%
- AOS (Acquisition of Signal) Time: The time in UTC when the satellite will be over the horizon and a signal might be received from the satellite.
- Maximum Elevation: The highest elevation in degrees during the pass, 15 degrees.

Based on this information, pick a satellite to schedule an observation. Here are some guidelines:

- You will be given a time range in UTC for choosing your pass. This will give time for the observation to be scheduled and avoid conflicts with other classes.
- For a better chance of getting a Successful Observation, pick a pass where the Maximum Elevation is more than 25 degrees.
- A satellite with a higher Success Rate might be more likely to give a successful observation so you might want to choose one with greater than 50%. On the other hand, if you get a successful observation of a satellite with a low rate, that could be an exciting result.

Enter your chosen satellite Name, Your Name, and AOS time in the class Google Doc. Make sure no one else in the class has selected the same satellite pass. You can select the same satellite as long as it is a different pass (the AOS time is different).

We will discuss the chosen satellite passes in class, then move on the next step.

2. Choose a Transmitter/Encoding for your Satellite (10 mins)

A satellite can have a number of different transmitters, each sending signals using different encoding and modulation. For your chosen satellite, you will need to choose one of the Encodings for your observation. To find out what transmitters/encodings are available for your satellite, click on your satellite name:



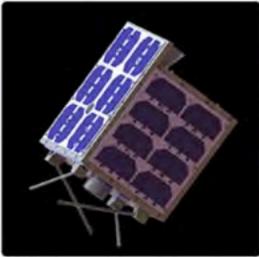
The list shows the different Transmitters for the satellite. In this case, it is CW Beacon (Morse Code), FM Transponder (analog voice), APRS Digipeater (digital packet radio signals), and GFSK Telemetry (Gaussian Frequency Shift Keying digital telemetry). Your satellite might have a different list or only one. The success rate for each one is shown by the bars. For instance, in this example, the APRS Digipeater has a relatively low success rate while the GFSK Telemetry has the highest number of observations and the highest success rate.

Click on the “SatNOGS DB” button in the window to get the Transmitter Database.

Look at the map to see where your satellite is currently located. In this case, LilacSat 2 is off the coast of Argentina at the time of this screenshot.

Enter the Location in the Google Doc as the nearest country or body of water or landmark.

40908 - LILACSAT 2 CAS-3H XW-2H alive Observations



Built by students at Harbin Institute of Technology (HIT), it carries 4 payloads. A VHF / UHF Software Defined Radio (SDR) FM repeater and APRS digipeater as well as CW telemetry and beacon, A second AIS / ADS-B receiver, An FPGA testing platform and a thermal infrared camera.



CW Beacon 🔗 ⓘ

Type: Transmitter

Service: Unknown

Downlink Mode: CW

Downlink: 437.200 MHz

Updated on 2019-04-18 05:39 by None Citation

FM Transponder 🔗 ⓘ

Type: Transceiver

Service: Unknown

Downlink Mode: FM

Uplink Mode: FM

Downlink: 437.200 MHz

Uplink: 144.350 MHz

Updated on 2019-04-18 05:39 by None Citation

APRS Digipeater 🔗 ⓘ

Type: Transceiver

Service: Unknown

Downlink Mode: AFSK

Uplink Mode: AFSK

Baud: 1200.0

Downlink: 144.390 MHz

Uplink: 144.390 MHz

Updated on 2019-04-18 05:39 by None Citation

GFSK Telemetry 🔗 ⓘ

Type: Transmitter

Service: Unknown

Downlink Mode: GFSK

Baud: 4800.0

Downlink: 437.225 MHz

Updated on 2019-04-18 05:39 by None Citation

BPSK Telemetry 🔗 ⓘ

Type: Transmitter

Service: Amateur

Downlink Mode: BPSK

Baud: 9600.0

Downlink: 437.200 MHz

Updated on 2020-03-18 10:58 by pierros Citation

You will see the same list of Transmitters on the satellite as before, but with more information. The Downlink is the frequency of the signal transmitted by the satellite to earth. Some Transmitters also have an Uplink frequency. This is the frequency that Amateur Radio operators can transmit on to send signals to the satellite. Our satellite observations in this activity are signals received from the satellite, so we are only concerned about the Downlink frequency.

The Villanova SatNOGS station has a VHF (Very High Frequency) Lindenblad antenna which can only receive signals in the frequency range 135 MHz to 148 MHz. You can only choose a

Transmitter that has a Downlink frequency in the range of 135 – 148 MHz. In this case, only one of the Transmitters is in this range, the APRS Digipeater at a frequency of 144.390 MHz. If there are multiple ones, then you have a choice.

Choose one that you are interested in that has a high success rate and a large number of observations (the previous window has the information). From the previous window, GFSK Telemetry has the highest success rate, but that Transmitter is at 437.225 MHz, which is in the UHF (Ultra High Frequency) band, outside of the range of our antenna.

Enter your chosen Transmitter Name, Downlink Mode, and Downlink Frequency in the class Google Doc.

We will discuss the chosen Transmitters in class before moving on the next step.

3. Identify a Successful Observation of your satellite and Transmitter (10 mins)

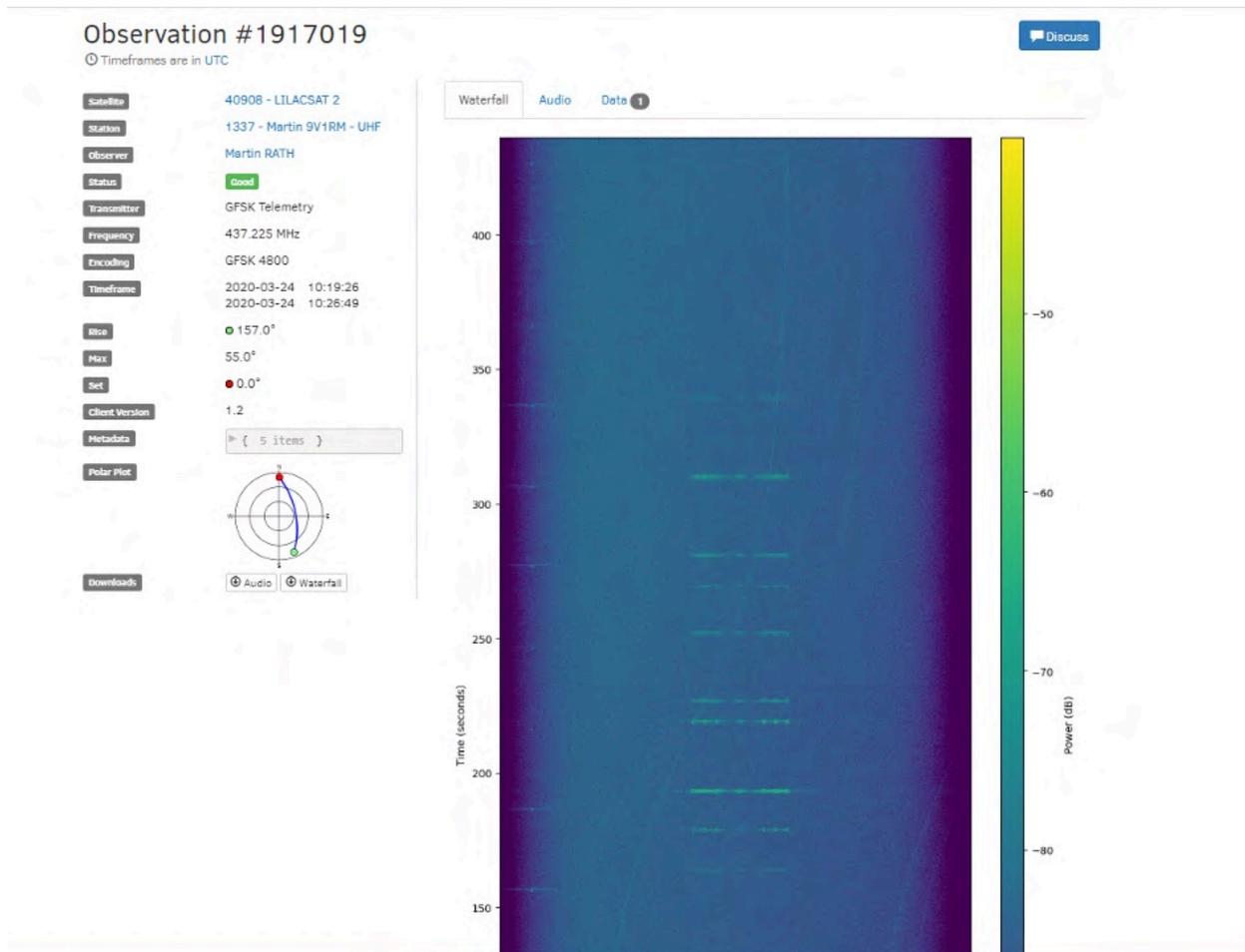
For your chosen satellite, you will search for a successful observation in a previous pass. Do this by clicking on your satellite name as in the previous step. Under Observations, there will be a number in a green box – click on that button to open a list of Successful Observations.



In this list, the different Transmitters are listed as “Encodings”. If there are multiple Transmitters, they will be mixed together in this list, so you will need to look for Successful Observations (green) that list your chosen Transmitter (Encoding). Right click on a few to open them in a new tab and check to see if they are in fact good observations (some Bad observations are incorrectly classified as Good). For example, here is one Successful Observation is shown below.

The observation appears to be good because there is a signal in the middle of the waterfall (light green) which is a digital signal (a series of lines). And the signals are roughly in a straight line (not curved) indicating that the signal matched the expected Doppler frequency shift.

Once you have found a Successful Observation, copy the link (URL) into the Google Doc.



4. Try to find a Successful Observation with Decoded Data (10 mins)

A successful observation with decoded data will have a number next to the observation in a list, such as this:



In the actual observation page, there will be a number in the Data tab, such as the previous observation has "Data(1)". Click on the Data tab to see the data:

Sometimes, the data will be an image, depending on the satellite. In this case, it is raw hexadecimal numbers. Using software or a spreadsheet, we could decode this telemetry to see the information, which could be temperature, current, and voltage information about the satellite. We will do this in a future class activity.

If you can find an observation with decoded data, add it to the Google Doc. If you can't, that's OK – not every satellite has decodable data.

In the next class period, we will check on your observation. You will decide if the observation is Good or Bad.

Appendix B - CubeSatSim Telemetry Activity

Before class students:

Pair up with their antenna partner. Hand out one RTL-SDR to each pair.

- In each pair, one needs to run Zadig and install FoxTelem
- Run FoxTelem and select Source RTL-SDR and frequency 434840 mode DUV/FSK and click Start
- Monitor audio

- If it works, they are ready
- If they have run FoxTelem before, clear stored packets

Before class instructor:

- Make sure CubeSatSim is charged up
- Put in CubeSatSim in APRS mode (1 blink)
- Shutdown

In Class:

Launch

Announce we are going to finally “launch” the CubeSatSim. The students will be in charge of receiving telemetry from the satellite and figuring out what is going on.

Instructor:

- Remove the RBF
- Put on cart on turntable with +Z axis up on L speed in front of LED lamp (need power cable)
- Wheel cart into hallway out of sight of students.
- Start up Raspberry Pi Ground Station and share screen with students.

First Contact

Announce: after the launch, the CubeSat is flying over our ground station for the first time. We need to make first contact and see if everything is “nominal” - functioning normally.

- Students run FoxTelem and listen for signal
- Do you see the signal? (Yes, sometimes a burst of signal)
- Can you decode? (No frames decoded by FoxTelem)
- Why not? (APRS packets not DUV/FSK telemetry)

Announce we have received new info from the software development team - if the CubeSat has had a power failure, it can start in random mode.

Announce we will send a command to the CubeSat to change to FSK/DUV mode (TA goes into the hall and changes it to two blinks)

- Now can you decode? (Yes)
- What is Battery voltage? Is that good? (3.8 V and above is good)

- What is Temperature? Is that good? (room temperature is fine)
- Are you getting any voltage or current from solar panels? Is that good? (around 4V is good)

Announce that we need to determine the rate of rotation using two different methods.

- A. Period of rotation using solar panels current peaks
- B. Rotation rate from gyro/IMU

Assign half the class to use method A and the other half to measure method B.

Write down results of A and B on the board. Are they the same?

Announce that we need to determine the axis of rotation using two different methods.

- A. Solar panel X Y Z current peaks order and relative height
- B. Relative values of rotation of X Y Z from gyro/IMU

Write down results of A and B on the board. Ask are they the same?

Announce that each orbit the rotation can be different.

TA puts on side (+X facing up) and changes turntable speed to H

What is the new axis of rotation?

What is the new rotation rate?

Announce it is another orbit later.

TA puts 50 Ohm terminator in place of antenna and puts a black postit on +Z and -Z solar panels.

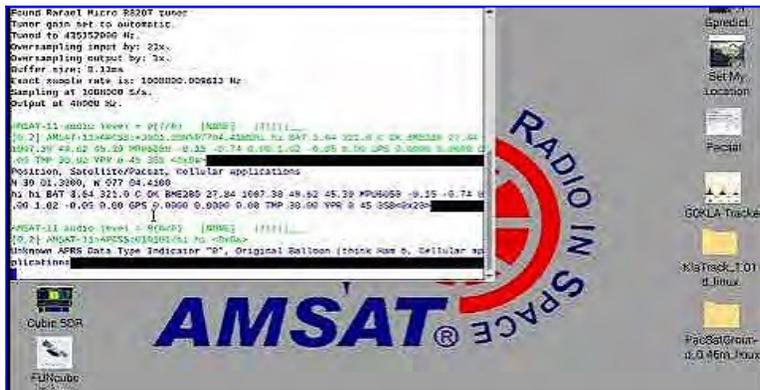
Ask if anything has changed?

- Answer: Yes, signal weaker
- Answer: Yes, solar panel failures

If time, have TA switch to SSTV (four blinks). Stop turntable and put cartoon alien photo in front of camera.

Decode using speaker (a battery powered speaker works well) and phone app (SSTV or Robot36). Also show Raspberry Pi station, too.

CubeSatSim YouTube Playlist Links



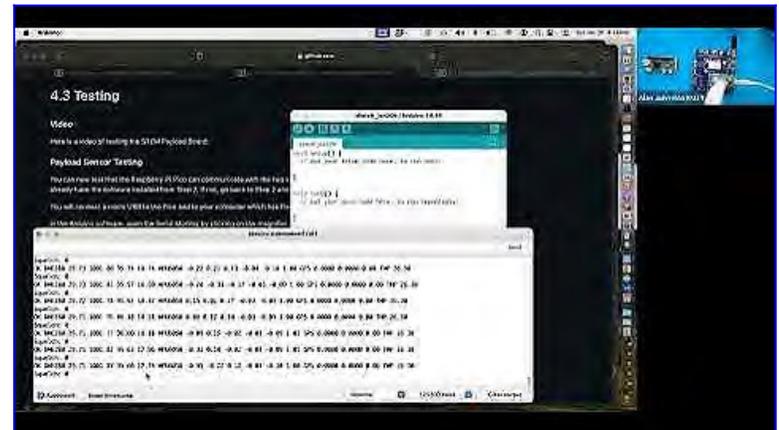
[AMSAT Fox in a Box version 4](#) [View full playlist](#) [1 video](#)



[CubeSatSim Kit Videos](#) [View full playlist](#) [5 videos](#)



[CubeSatSim Build Videos v2](#) [View full playlist](#) [12 videos](#)

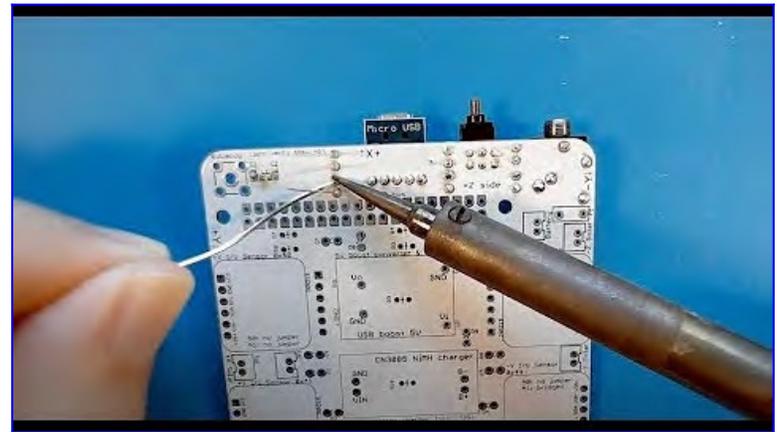


[CubeSatSim Beta Build v1.3.2](#) [View full playlist](#) [8 videos](#)

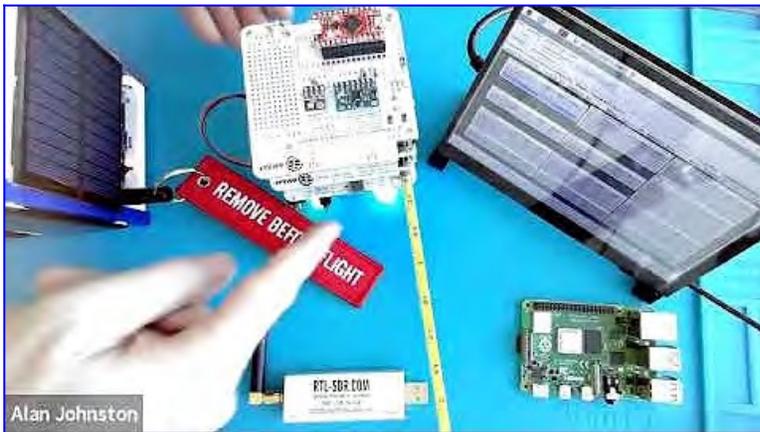
CubeSatSim YouTube Playlist Links



[AMSAT CubeSatSim Beta Kit Videos](#) [View full playlist](#) **1 video**



[CubeSatSim Build Videos v1](#) [View full playlist](#) **16 videos**



[CubeSatSim Quick Start Guides](#) [View full playlist](#) **1 video**



[AMSAT Fox-in-a-box Version 3](#) [View full playlist](#) **3 videos**

CubeSatSim YouTube Playlist Links



[AMSAT CubeSatSim Lite View full playlist](#) [7 videos](#)



[ARISS Radio Pi](#) [View full playlist](#) [4 videos](#)

AMSAT CubeSatSim Glossary

Here's a handy glossary for the various technical terms and acronyms associated with Amateur Radio and the CubeSatSim.

Term	Meaning
AFSK	Audio Frequency Shift Keying.
Amateur Radio	A popular technical hobby and volunteer public service that uses designated radio frequencies for non-commercial exchange of messages, wireless experimentation, self-training, and emergency communications. The CubeSatSim transmits telemetry on frequencies within the Amateur Radio bands.
Amp	Short hand for Ampere, which is the SI base unit of electrical current.
<u>AMSAT</u>	The Radio Amateur Satellite Corporation (AMSAT) is a non-profit volunteer organization which designs, builds, and operates experimental satellites and promotes space education. We work in partnership with government, industry, educational institutions, and fellow Amateur Radio societies. We encourage technical and scientific innovation and promote the training and development of skilled satellite and ground system designers and operators. AMSAT is a worldwide group of Amateur Radio operators who share an active interest in building, launching and then communicating with each other through non-commercial Amateur Radio satellites. There are similar independent AMSAT organizations around the world.
Antenna	A device that converts an electrical signal into a radio wave, or a radio wave into an electrical signal for use in a radio system. Typical antennas are designed for use on specified radio frequencies.
Antenna, Handheld	Often used in combination with a handy talkie (HT) or larger mobile radio to receive and transmit signals to and from orbiting Amateur-Satellites. Mostly directional, popular manufacturers include Arrow Antennas and Elk Antennas among others. Students sometimes construct a directional antenna using <u>metal tape measure</u> for the antenna elements mounted on a wood or plastic boom approximately 3-5 feet in length. The antenna is connected to the radio using coaxial cable.
APRS	Automatic Packet Reporting System. APRS is an Amateur Radio-based system for real time digital communications of information that includes GPS coordinates. APRS data can be displayed on a map.
Arduino	A small single board computer. The Arduino Integrated Development Environment (IDE) is used to compile software used in the Raspberry Pi Pico for the CubeSatSim project.
Beacon	Is the satellite's signal that signifies its presence to the world and can be tracked by almost any ground station or Amateur Radio operator. Generally, the signal in beacon primarily has the call sign of the satellite and may also have health data or telemetry data.
BME280	A sensor used to measure relative humidity, barometric pressure and ambient temperature. The BME280 sensor uses the I2C communications interface bus in the CubeSatSim.

Term	Meaning
bps	The data transmission rate in Bits Per Second (bps).
BPSK	Binary Phase-Shift Keying. A digital modulation scheme that conveys data by changing, or modulating, two different phases of a reference signal.
Bus	A conductor, or group of conductors, that serve as a common connection for two or more circuits.
Calibration	The process of converting the numerical measurement of a sensor to a known value. After calibration, a sensor gives more accurate readings.
Callsign	A combination of letters and numbers used to identify a radio station. Used by amateur radio operators and also commercial radio and TV transmitters.
Coax	Short form of Coaxial cable. A special form of dual-conductor radio frequency (RF) cable with a center conductor wire insulated from an outer mesh wire or other material forming the ground. RF cables used in amateur transceivers and receive only radios like SDRs.
Command and Control (C²)	Satellite Command and Control (C2C) messages are used to connect, usually over radio, with a satellite and upload commands, settings, schedule operations and experiments, and other functions.
CPU	The Central Processing Unit (CPU) is the "brains" of a computer that executes instructions and controls input and output. Also just known as a processor.
CubeSat	Is a small satellite with a form factor of 10 cm cubed. A 1U CubeSat has a mass of no more than 2 kg, and often use commercial off-the-shelf (COTS) components for their electronics and structure.
CubeSatSim	The AMSAT CubeSat Simulator is a tool for education and demonstrations. It can be used in a classroom or training setting to introduce the basics of nanosatellites, or it can be used to teach STEM (Science Technology Engineering and Math) concepts, or it can be a stepping stone in a project to build and launch an actual CubeSat into space.
Current	Electrical current is the flow of charged particles, usually electrons, measured in the units of Amperes or Amps or milli Amps.
CW	Continuous Wave. AKA Morse Code. The very first digital mode of communication.
Demodulation	Is the process of extracting information from a transmitted signal.
Diode	Is an electrical element that only allows current to flow in one direction.
Direwolf	A popular open source computer software program used to decode APRS packet and DTMF signals, among others.
Downlink	Radio communication from a satellite to a ground station. Uplink is radio communication in the other direction.
Drivers	software installed on a computer to allow a device to be accessed by software on the computer. For example, an RTL-SDR radio dongle needs to have drivers installed on a computer to be used.

Term	Meaning
DTMF	Dual Tone Multi Frequency (DTMF) tones are used over the radio or network to send digits or letters. They have been used over telephone networks in the past. In the CubeSatSim, they can be used for Command and Control signals.
Eye Diagram	A display in which a digital signal from a receiver is repetitively sampled and applied to the vertical input (y-axis), while the data rate is used to trigger the horizontal sweep (x-axis). Eye diagrams are displayed and utilized by the FoxTelem software when the CubeSatSim is transmitting in the FSK or BPSK mode.
FCC Part 97	Rules and regulations related to the Amateur Radio Service, found in Section 97.3 of the U.S. Code of Federal Regulations.. These definitions include Amateur operator, the licensed person responsible for a station; Amateur service, a radio communication service for non-commercial, personal aims; Amateur station, the apparatus for radio communications; and concepts like Call sign, Control operator, Remote control, Space station, and Telecommand station.
FFT	Fast Fourier Transform (FFT) is a mathematical method to calculate and display a radio spectrum in a graphical way.
FIAB	Fox-in-a-Box (Video) A small, easily-deployed Fox satellite telemetry receiver (ground station) that utilizes a Raspberry Pi and an SDR dongle. FIAB can be used as the Ground Station for the CubeSatSim. A Raspberry Pi image file is available for download or purchase pre-installed on a SD card from AMSAT.
Fox-1	The Series of five 1U CubeSats identified by a letter (A through E) launched into Low Earth Orbit by AMSAT. The FoxTelem software decodes the telemetry transmitted by these satellites.
FoxTelem	A software program included with the Fox-In-a-Box ground station image file, used to decode Fox-1 FSK and BPSK CubeSatSim telemetry.
FSK	Frequency Shift Keying. This is the digital modulation technique in which the frequency of the carrier signal varies according to the digital signal changes. FSK is a type of frequency modulation.
FUNcube	A 1U CubeSat launched into space by AMSAT-UK also known as AO-73 that transmits telemetry in BPSK 1200 bps.
Gain of an Antenna	The ratio, usually expressed in decibels, of the power required at the input of a loss free reference antenna to the power supplied to the input of the given antenna to produce, in a given direction, the same field strength or the same power flux-density at the same distance. When not specified otherwise, the gain refers to the direction of maximum radiation. The gain may be considered for a specified polarization.
GPIO	General Purpose Input Output. In the context of the CubeSatSim, this is the 40-pin header used on the Raspberry Pi Zero and Raspberry Pi 4. The GPIO pins on the Raspberry Pi Pico are along the outside edge of the Pico's printed circuit board. These GPIO pins are the physical interface between the Raspberry Pi and the CubeSatSim sensors and peripheral devices.

Term	Meaning
GPS	Global Positioning System is a satellite-based navigation system owned by the United States Space Force. It is one of the global navigation satellite systems (GNSS) that provide geolocation and time information to a GPS receiver anywhere on or near the Earth where signal quality permits. Also known as a Radio navigation Satellite Service (RNSS). The GPS project was started by the U.S. Department of Defense in 1973. The first prototype spacecraft was launched in 1978 and the full constellation of 24 satellites became operational in 1993. Over time GPS receivers were miniaturized to tiny microcircuits and incorporated into an extremely wide range of devices.
Ground Station	Used to communicate and command satellites in space. (See AMSAT Journal Vol 42, No. 4 for a detailed description of the CubeSatSim ground station).
gyroscope	A gyroscope or just a gyro is a device that measures rotation rate. The MPU6050 sensor in the CubeSatSim has a gyroscope.
HAB	High Altitude Balloon. High altitude ballooning is a hobby that allows hobbyist to explore earth's upper atmosphere utilizing customized payloads.
Ham Radio	A slang term for The Amateur Radio Service.
HT	A Handy Talkie (HT) is a term for a portable battery powered radio ham radio transmitter and receiver that you can hold in your hand. These devices range in price from under \$50 to about \$800 depending on quality and sophistication.
I2C	Inter Integrated Circuit bus. A synchronous, multi-master/multi-slave, single-ended, serial communication bus used to communicate between the Raspberry Pi Zero and various sensors in the CubeSatSim.
IDE	Integrated Development Environment. An IDE is a software application that helps programmers develop software code efficiently.
IHU	Internal Housekeeping Unit. Housekeeping data includes items like temperature, voltage and attitude.
Image File	Is a file that contains a complete copy of another drive or SD Card. Also known as a disk image. An image file containing common ground station software is available for the Raspberry Pi. The image can be downloaded and flash onto an SD Card for quickly setting up a CubeSatSim ground station.
IMU	Internal Measurement Unit. An IMU often contains an accelerometer and gyroscope.
INA219	A sensor for measuring current and voltage. In the CubeSatSim, the INA219 sensors are connected to the Raspberry Pi Zero via the Pi's I2C bus.
ISM Band	Portions of the radio spectrum reserved internationally for Industrial, Scientific, and Medical (ISM) purposes.
kHz	Kilo Hertz (kHz) is a radio frequency in thousands of cycles per second.
LED	Light Emitting Diode.
LEO	Low Earth Orbit. LEO is an orbit that is relatively close to Earth's surface. It is communally considered an altitude of less than 1000 km above Earth.

Term	Meaning
MHz	Mega Hertz (MHz) is a radio frequency in millions of cycles per second.
Mode	A Mode can be defined as how something is done or how it happens. The CubeSatSim uses five different forms of modulation, or modes, to send information to the ground stations. These modes are CW, FSK, BPSK, APRS, and SSTV.
Modulation	Is the process of encoding information in a transmitted signal.
MPU6050	Is an IMU sensor that combines a gyroscope and accelerometer and uses a standard I2C bus for data communication.
Nickel Metal Hydride	Is a type of rechargeable battery used in the CubeSatSim. Also written as NiMH, these batteries are safer to use than Lithium batteries. Abbreviated NiMH or Ni-MH.
Open Circuit	A open circuit (sometimes written as o/c or oc) is when nothing is connected to an electrical output. For example, the open circuit voltage of a solar panel is the maximum voltage that can be measured when nothing is connected between the plus and minus wires of the solar panel.
OSCAR	Orbiting Satellite Carrying Amateur Radio.
Packet	A small block of data transmitted together over a radio or computer network.
Payload	A payload is made up of scientific instruments and all the electronics that support a given mission. The CubeSatSim simulates a satellite payload and can be used as a high altitude balloon payload.
Phasor diagram	A graphical representation, plotted on a coordinate system, of the phase relationship of the two phases of a BPSK modulated signal sent from the CubeSatSim. It shows the transitions from one phase state to the other and is displayed when decoding BPSK in the FoxTelm software.
Pi Zero	A Pi Zero is one of the smallest Raspberry Pi Single Board Computers (SBC) that is used as the processor in the CubeSatSim.
Pitch	One the principle axes used in aviation and aerospace that refers to the up or down angle of the nose of an aircraft or spacecraft. (Also see yaw and roll)
Potentiometer	A Potentiometer is a variable resistor who resistance can be changed by turning a screw or moving a slide. Also known as a Trimpot or just a Pot.
Power	Electrical Power is the rate of doing work or using energy, and is measured in Watts or milli Watts. Electrical power is calculated as the product of the Voltage and the Current.
PSU	Power Supply Unit. The battery board functions as the PSU in the CubeSatSim.
QSSTV	A software program included with the Fox-In-a-Box ground station image file, used to decode CubeSatSim Slow Scan TV pictures.

Term	Meaning
Raspberry Pi	Is a single-board computer made by the Raspberry Pi Foundation , a UK charity that aims to educate people in computing and create easier access to computing education. Two different form factors of the Raspberry Pi are used in the CubeSatSim (Raspberry Pi Zero and Raspberry Pi Pico).
RBF	Remove Before Flight. For the CubeSatSim, the RBF switch activates the CubeSatSim and allows operation using internal battery power.
Repeater	A radio repeater receives a radio signal on one frequency and retransmits the signal on a different repeater. A cross-band repeater receives on one radio band and retransmits on a different radio band.
RF	Radio Frequency. Frequency is usually expressed in cycles per second. See kHz, MHz and GHz
Roll	One the principle axes used in aviation and aerospace that refers to the up or down angle of the wings of an aircraft or the rotation of spacecraft about its path of flight. (Also see yaw and pitch)
RSSI	Received Signal Strength Indicator.
RTL-SDR	A popular low cost SDR with a USB interface that plugs into a computer such as a Raspberry Pi or laptop to receive radio signals.
SDR	Software Defined Radio. A SDR is a radio communication system that uses software for the modulation and demodulation of radio signals. (Also see RTL-SDR)
Sensor	A device which detects or measures a physical property and records, indicates, or otherwise responds to the physical property.
Short Circuit	A short circuit (sometimes written as s/c or sc) is a path of minimal resistance. For example, the short circuit current of a solar panel is the maximum current that can be measured when a wire is connected between the plus and minus wires of the solar panel.
SI unit	International System of Units are the standard modern form of the metric system, used globally for scientific and everyday measurement.
SNR	Signal-To-Noise ratio. SNR is the ratio between the desired information or the power of a signal and the undesired signal or the power of the background noise.
Software-Defined Radio	A device or system enabling computers to tune a wide spectrum of radio frequencies, using software instead of hardware for modulation and demodulation
Solar Panel	A device that coverts energy from sunlight into electrical energy to run a CubeSat or charge batteries.
Space Communications	See AAS Chapter 23 . PDF via Google Scholar.
Space Tracking	Determination of the orbit, velocity or instantaneous position of an object in space by means of radio determination, excluding primary radar, for the purpose of following the movement of the object.

Term	Meaning
SSTV	Slow Scan Television. SSTV is a way of sending pictures over a voice bandwidth channel using audio tones.
Telecommand	The use of radio communication for the transmission of signals to a space station to initiate, modify or terminate functions of equipment on a space object, including the space station.
Telemetry	The process of recording and transmitting the readings of an instrument or sensor.
TLEs	Two-line Elements are a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time. This information is used to track satellites. Knowing the path of a satellite is very important for antenna pointing when attempting satellite radio communications.
Transceiver	Is a combination of a radio transmitter and radio receiver.
Troubleshooting	Is the method of trying to determine the cause of a problem or anomaly or failure. Troubleshooting a problem with a satellite in space is usually done by a careful analysis of telemetry data.
UHF	Ultra-High Frequency. UHF radio frequencies range between 300 MHz and 3 GHz. The CubeSatSim transmits and receives in the UHF range of radio frequencies.
Uplink	Radio communication from a ground station to a satellite. Downlink is radio communication in the other direction.
USB	The Universal Serial Bus (USB) interface common on computers for charging and data communications. There are different types of connectors including micro USB, USB-C, etc. In relation to communication mode it may refer to Upper Side Band.
Volt	The SI unit of electromotive force, the difference of potential that would drive one ampere of current against one ohm resistance.
Waterfall display	Is a graphical representation of the signals across a frequency range, generally color-coded to indicate signal amplitude or strength, displayed over time.
Web SDR	A software program included with the Fox-In-a-Box ground station image file, can be used to decode CubeSatSim Packet telemetry.
Yaw	One the principle axes used in aviation and aerospace that refers to the left or right angle of the nose of an aircraft or spacecraft. (Also see pitch and roll)



Strategic Satellite Objectives and Organization Goals

Highly Elliptical Orbit

1. Upward to HEO. Develop and deploy a series of spacecraft capable of providing wide-area and continuous coverage from high-Earth and geostationary transfer orbits.

Greater Orbit, Larger Footprint

2 GOLF. Develop and deploy a series of increasingly capable spacecraft through a program to learn skills and systems for which we do not yet have the necessary low-risk experience, including active attitude control, deployable/steerable solar panels, radiation tolerance for Commercial off the Shelf (COTS) components in higher orbits, and propulsion.

Amateur Radio on the International Space Station

3. AREx-A. Partner with ARISS and ARISS-USA to advance Amateur Radio's presence aboard NASA's International Space Stations, Deep Space Gateway and Artemis missions and provide opportunities to engage with astronauts in lunar and deep space operations.

Low Earth Orbit

4. LEO. Support a stream of LEO satellites developed in cooperation with the educational community and other Amateur Radio satellite groups.

4.1 FM Operations. Develop, deploy, and support a series of 1u spacecraft to support continued FM amateur satellite operations in low Earth orbit.

4.2 Partnerships. Develop a plug-and-play communications solution for educational and other Amateur Radio CubeSat programs, providing a VHF/UHF telemetry beacon, command receiver, and linear transponder or FM repeater communications module.

AMSAT STEM Initiatives

5. AMSAT Education. Support science, technology, engineering, and mathematics (STEM) initiatives and training programs for satellite and ground system designers and operators.

5.1 CubeSat Simulator. Continue development of AMSAT's CubeSat Simulator Program.

5.2 High Altitude Ballooning. Develop program to support and sponsor the use of amateur radio in high-altitude balloon (HAB) launches.

5.3 Youth Initiative. Develop an educational outreach program that encourages youth to pursue STEM interests in space science and communication technology.

There's No Better Time to Join AMSAT than Right Now!

For over 50 years the Radio Amateur Satellite Corporation (as AMSAT is officially known) has played a key role in significantly advancing the state of the art in space science, space education, and space technology. By becoming a member, you can help propel AMSAT into the next 50 years of discovery, adventure and excitement.

Your membership helps support AMSAT activities such as:

- Working in partnership with government, industry, educational institutions, and fellow amateur radio societies to foster Amateur Radio's participation in space research and communication,
- Designing and building communication satellites to operate in the harsh environment of space,
- managing those satellites, once they are in orbit, and ensure they are available for public use,
- Promoting space education and bringing space-based STEM initiatives to life,
- Providing opportunities for students to speak with astronauts aboard the International Space Station (ISS), and
- Collecting, compiling, and sharing what is learned with not only AMSAT members, but, with the world!
- Resources open to all Radio Amateurs such as the
 - Voice and digital communications via AMSAT-built satellites
 - AMSAT Website
 - AMSAT Bulletin Board
 - AMSAT News Service
 - Satellite status and orbital elements
 - And much, much more!

Plus, your annual or lifetime membership includes the following exclusive benefits:

- One-year subscription to *The AMSAT Journal*
- Access to past issues of *The AMSAT Journal* and Symposium Proceedings
- Discounted price on certain items in the AMSAT Store
- Opportunity to serve on committees and gain valuable professional experience
- Be nominated for awards
- Voting rights in elections
- Opportunity to run for office

Your membership in AMSAT is more important than ever and it takes only a few moments to join.

Do it *right now* while you're thinking about it! Go to launch.amsat.org and click on "JOIN US NOW." Your membership lasts for one year from the date of purchase.

Radio Amateur Satellite Corporation
712 H Street N.E. Suite 1653
Washington, DC 20002
info@amsat.org

