Overview of the Satellite Networked Open Ground Stations (SatNOGS) Project

Dan White
Valparaiso University
1900 Chapel Dr., Valparaiso, IN 46383; 219-464-6534
dan.white@valpo.edu

Corey Shields, Pierros Papadeas, Agisilaos Zisimatos, Manolis Surligas, Matthaios Papamatthaiou, Dimitrios Papadeas, Eleytherios Kosmas, Vasileios Tsiligiannis, Alexandru Csete, Nikolaos Roussos, Alfredos-Panagiotis Damkalis, Konstantinos Triantafyllakis, Georgios Vardakis
Libre Space Foundation
Ampatiellou 11, Athens 11144, Greece
info@libre.space

ABSTRACT

The SatNOGS, or Satellite Network Open Ground Stations, project promotes and supports free and open space applications. It seeks to address the problem of connecting many satellite users/observers to many ground station operators. Modern Open Source software, web, and hardware techniques are used in implementing the Network, Database, Client, and Ground Station sub-projects. Modularity in all the systems promotes the dual-use of ground stations by not interfering with local operation while utilizing the great amount of time a civilian, non-commercial ground station would otherwise sit idle.

INTRODUCTION

Once an individual or organization builds a ground station for small satellite downlinks, especially if not a commercial venture, the hardware ends up sitting idle for a great majority of the time. This capability, when not being actively used by the local owner, could be utilized for reception of other satellites. The SETI@home project is one of the earliest and well-known projects to make use of idle resources – compute cycles in that case.[1]

Recent advances in low-cost, software-defined radio (SDR) technology and 3D printing have put ground station ownership within the reach of individuals. Largely composed of Amateur Radio operators, these people receive telemetry and data from many satellites and provide the information to the owners and the general public.

What is missing is a civilian infrastructure to connect these many owners and ground station operators in a way that is flexible and open. The ESA's Global Educational Network for Satellite Operations (GENSO)[2] was a notable attempt at such a network aimed at University-class projects and stations. The fact that the `genso.org` domain name does not resolve to a live server is a practical indication of the current state of the project. Services, such as RBC Signals[3] and KSAT,[4] are also addressing this issue from the commercial side. The FUNcube project is a prime example of a well-organized effort to receive and collect data from satellite for educational outreach.[5]

History

Members of *Hackerspace.gr*[6] in Athens Greece first proposed SatNOGS as a part of the 2014 International Space Apps Challenge's "Virtual Ground Station App – Global Crowdsourcing of CubeSats" Challenge.[7] It was later submitted as an entry to the 2014 Hackaday Prize, ultimately winning the grand prize of $\$196,418$.[8] The funds provided seed money to start the Libre Space Foundation,[9] a non-profit dedicated to supporting free and open source space and related projects. One of the first extension projects from this was a collaboration between the Libre Space Foundation and the University of Patras with the QB50 project on the 2U CubeSat UPSat.[10] The satellite, launched on April 18, 2017 and deployed from the ISS on May 18, 2017, was one of the first open source satellite designs.

A unique aspect of the project is its position bridging the Maker, Hacker, and Amateur Radio communities. The 2015 TAPR/AMSAT Banquet's speaker,

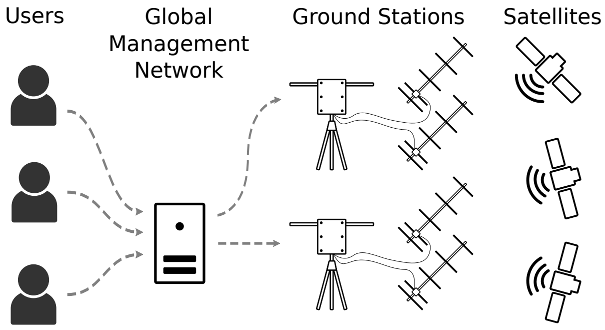


Figure 1: General architecture of SatNOGS interactions.

Figure 2: Four major software components of SatNOGS.

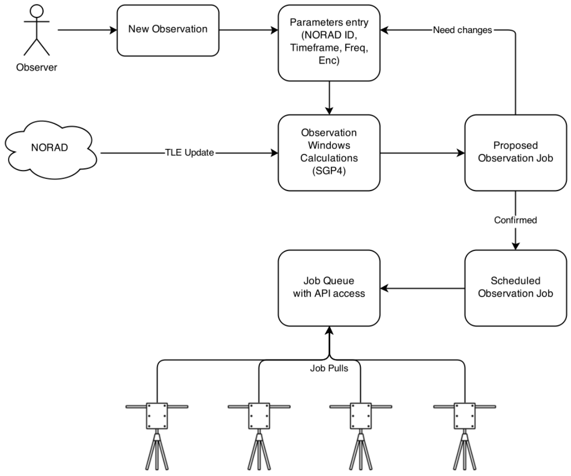

Observer
New Observation
Parameters entry (NORAD ID, Timeframe, Freq, Enc)
Need changes
NORAD
TLE Update
Observation Windows Calculations (SGP4)
Proposed Observation Job
Confirmed
Job Queue with API access
Scheduled Observation Job
Job Pulls

Figure 3: Network operations involved in scheduling an observation of a specific satellite by a certain ground station.

Michael Ossmann ADØNR of Great Scott Gadgets and creater of the HackRF, at the 2015 TAPR/AMSAT Banquet, pointed out the many characteristics valued and shared by these two communities.[11] Ward Silver, NØAX's article for Makezine also draws these connections.[12]

Overview

This paper gives an overview of the SatNOGS[13] project's major components. Figure 1 gives a high-level overview of the relationship between users and ground stations. Figure 2 illustrates the relationship of the four major sub-projects: Network, DB, Client, and Ground Station, to the other components of a ground station. Each sub-component is described in the following sections. The modular approach promotes re-use of existing components at a ground station.

All code and hardware designs for the project are publicly available under Open Software (AGPLv3 and GPLv3) and Open Hardware (CERN OHLv1.2) licenses.[14] Software documentation may be found at.[15] Additional information and tutorials are at the community-supported WiKi.[16] Data collected and stored in the Network and DB components are freely available under the Creative Commons Attribution-Share Alike license.[17]

NETWORK

The SatNOGS Network is accessed by users via a web interface. The user provides details about observation that they would like to schedule (which satellite, which band, time-frame, signal encoding, etc.). From this information, the system calculates the possible observation windows from the currently available Ground Stations connected to the Network having the necessary capabilities. Once the observer confirms the proposed "Observation Job" then it is sent as a job to each Ground Stations' job queue to be executed. Figure 3 diagrams this process.

The Network website provides a real-time count of the ground stations registered on the network. At the time of the writing, there are 33 online and 25 additional stations in testing mode. An additional staging server is available to allow testing of ground station client software before being made available for scheduling on the production network.[18]

Account holders on the Network site (owners of connected ground stations), are able to schedule observation jobs on any of the ground stations for any configured satellite. Users search for either desired satellite or ground station and the server computes pass information for matching observation windows. Then they select the desired passes and schedule the

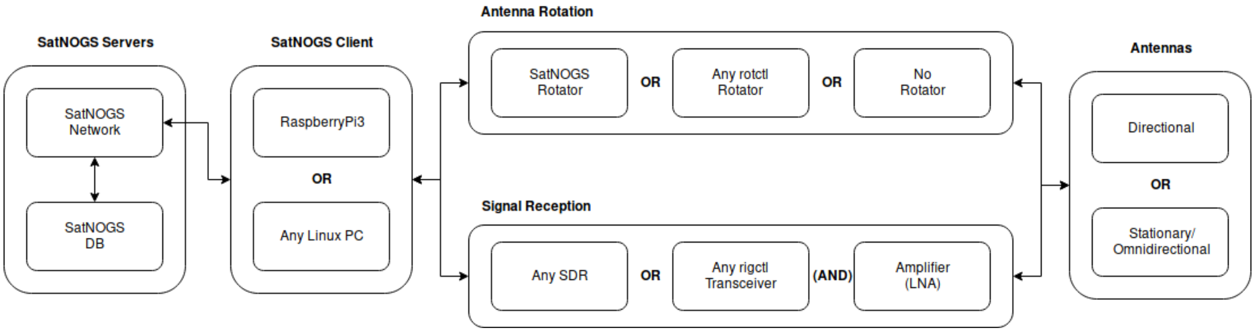


Figure 4: Interactions between the SatNOGS Network servers and a ground station.

jobs.

Ground Stations query the Network server for its current list of scheduled jobs, execute those observations, collect observation data, then upload it back to the Network through a REST API. Uploaded data is then made available to the initiating user and any other third party via the observation's ID. Modern web technologies are used on the Network website to provide timeline and recording visualization and playback directly in the browser. Download links are also provided for further offline analysis. No special software or installation is necessary for users to interact with the Network. SatNOGS Network provides an API to allow other applications and services to query information and, in the future, automatically schedule observations.

Figure 5 shows the metadata and waterfall plot that is recorded for a successful job. In this case, it is for the VHF FM downlink portion of an ARISS[19] school contact with an ISS astronaut, recorded as observation #130609. Also, Figure 6 shows the web view of the embedded audio player widget which allows listening to the demodulated downlink audio.

The Client component uploads a waterfall plot of the received spectrum, demodulated audio recording, and decoded data frames found in the recording. The audio recordings are then archived at the Internet Archive in the SatNOGS collection.[20] Figure 7 shows an example of the Network website's display for uploaded data-under-voice (DUV) frames from AMSAT-NA's AO-85 or FOX-1A CubeSat.[21]

DATABASE

The coordination and aggregation provided by the Network component requires a centralized source of satellite information such as frequencies and transmission modes. SatNOGS DB was created to address the fact that there was no known public source for this information in a machine-readable format. In the same spirit of the rest of the project, the database is open access and not specific only to the SatNOGS project. For example, the Android and iOS app ISS Detector uses SatNOGS DB for obtaining satellite information.[22]

Specifically, DB is a crowd-sourced suggestions app for transponder data. Satellites are identified by their NORAD space object catalog number and their common name. Each object has an associated set of transponder records which indicate frequencies and modulation formats. SatNOGS Network pulls this information when calculating possible observations. Orbital information is not stored in the DB component, and is updated from CelesTrak[23] and stored with each observation's metadata.

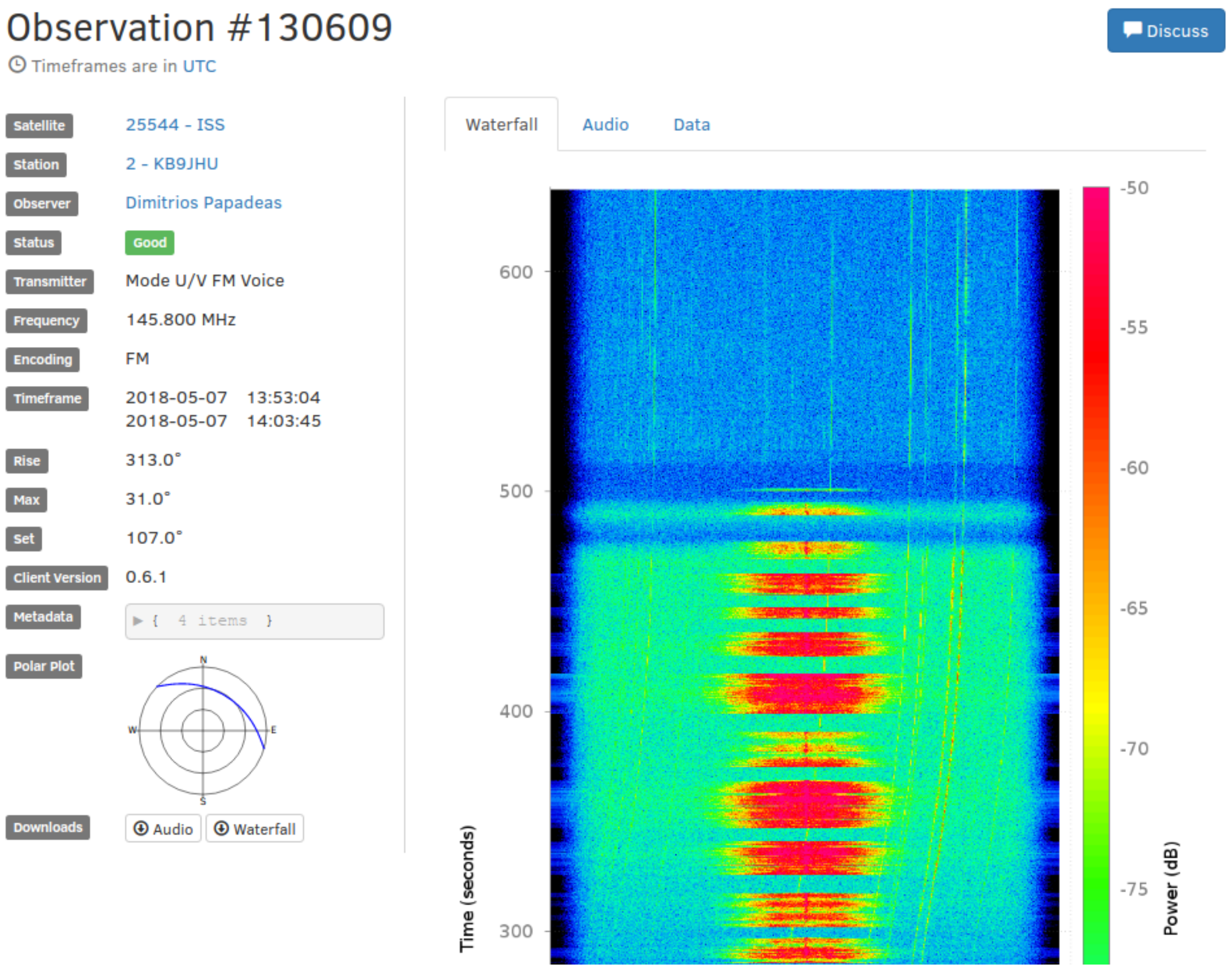
Updates to information in the database are encouraged and accepted from the public and also from other sources of satellite information with open APIs. As with the Network, user interaction is purely web-based and requires no additional software besides a modern web browser.

Besides satellite information, the DB component is a repository for collected telemetry data. The API works with third-party apps including the DK3WN TLM Forwarder? and `gr-satellites`.[24] The database currently stores information about 250 satellites with 433 transmitters and over 16 million telemetry frames.

CLIENT

SatNOGS Client consists of software running a computer which controls the ground station hardware. Figure 10 diagrams the internal, modular, components of the client's interaction with the Network.

The poller regularly checks the Network API for observation jobs scheduled for the local Ground Station. Information contained in each job includes satellite



Observation #130609

Timeframes are in UTC

Satellite	25544 - ISS
Station	2 - KB9JHU
Observer	Dimitrios Papadeas
Status	Good
Transmitter	Mode U/V FM Voice
Frequency	145.800 MHz
Encoding	FM
Timeframe	2018-05-07 13:53:04 2018-05-07 14:03:45
Rise	313.0°
Max	31.0°
Set	107.0°
Client Version	0.6.1
Metadata	▸ { 4 items }
Polar Plot	
Downloads	Audio Waterfall

Figure 5: Observation information and waterfall plot of signal received from ARISS contact.

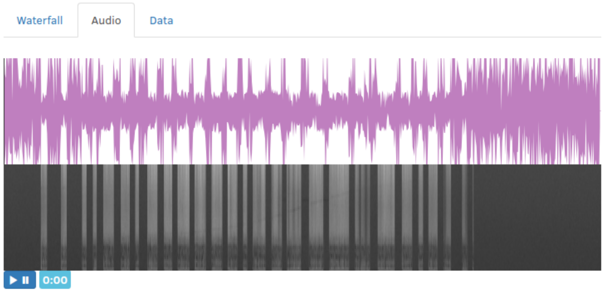
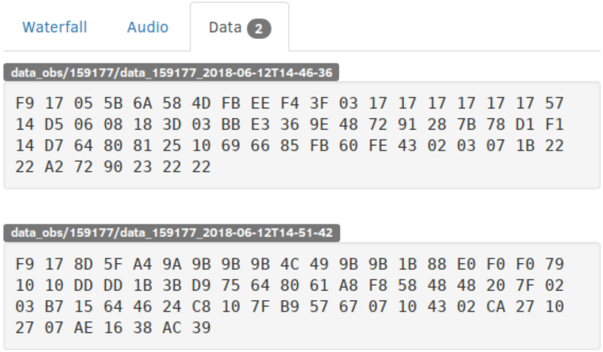


Figure 6: Demodulated audio player tab from observation #130609.



data_obs/159177/data_159177_2018-06-12T14-46-36

```
F9 17 05 5B 6A 58 4D FB EE F4 3F 03 17 17 17 17 17 17 57
14 D5 06 08 18 3D 03 BB E3 36 9E 48 72 91 28 7B 78 D1 F1
14 D7 64 80 81 25 10 69 66 85 FB 60 FE 43 02 03 07 1B 22
22 A2 72 90 23 22 22
```

data_obs/159177/data_159177_2018-06-12T14-51-42

```
F9 17 8D 5F A4 9A 9B 9B 9B 4C 49 9B 9B 1B 88 E0 F0 F0 79
10 10 DD DD 1B 3B D9 75 64 80 61 A8 F8 58 48 48 20 7F 02
03 B7 15 64 46 24 C8 10 7F B9 57 67 07 10 43 02 CA 27 10
27 07 AE 16 38 AC 39
```

Figure 7: Decoded DUV telemetry data from FOX-1A.

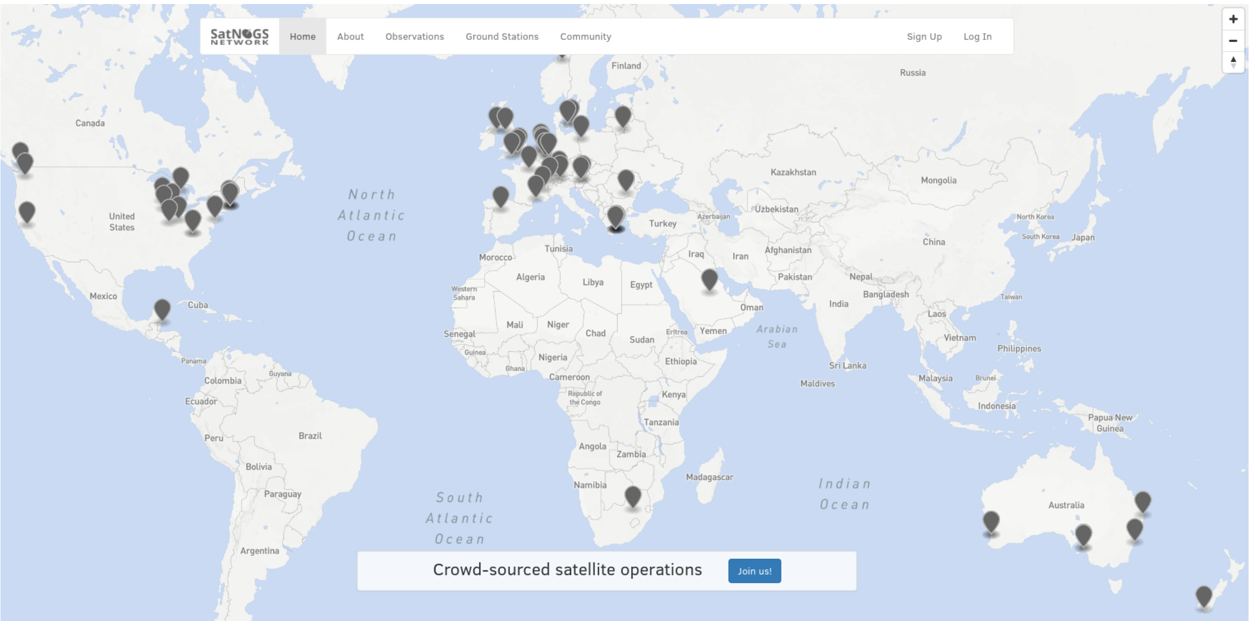
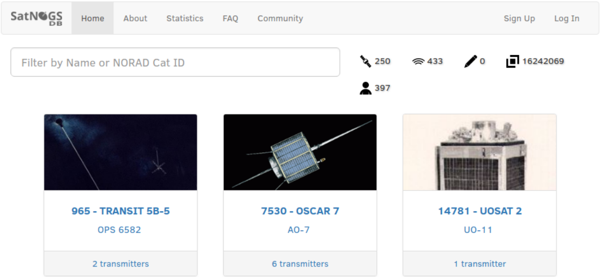


Figure 8: Active SatNOGS Ground Station locations.



965 - TRANSIT 5B-5
OPS 6582
2 transmitters

7530 - OSCAR 7
AO-7
6 transmitters

14781 - UOSAT 2
UO-11
1 transmitter

Figure 9: Screenshot of the SatNOGS DB front page.

orbital elements, receiver tuning and demodulation parameters, and timing. The PyEphem[25] library is used to calculate the necessary antenna pointing and relative velocity for doppler shift calculation during the pass.

Where possible, existing standard interfaces are used within the Client. These include using the daemons `rotctl` for rotator control and `rigctl` for frequency control from HamLib.[26] Any rotator controller which works with `rotctl` automatically works with the SatNOGS Client by providing the appropriate IP address and port number the rotator is listening on.

Control and operation of the receiver hardware and processing uses GNU Radio.[27] Flowgraphs from the out-of-tree module `gr-satnogs` contain the Doppler tuning, translation, demodulation, and decoding con-

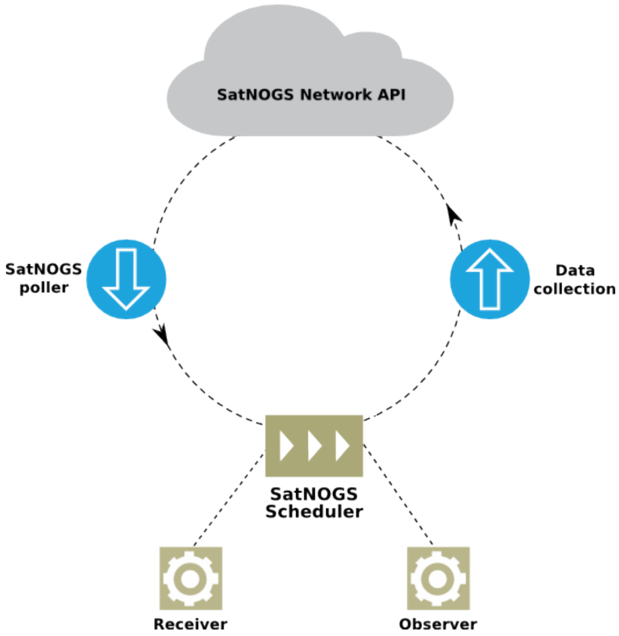


Figure 10: Client communication and operation modules.

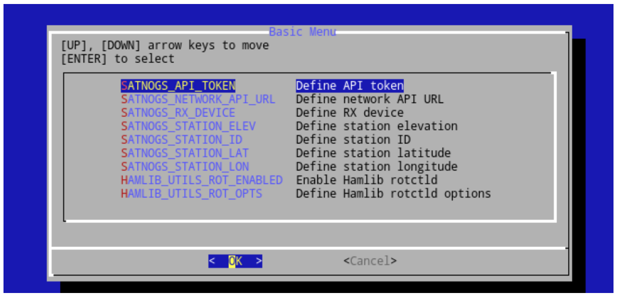
figuration for each of the *transmitter modes* as defined in the DB. These are either generic modulation and baud rate tags such as `BPSK1k2` or `GMSK9k6`, or satellite specific settings such as the AMSAT FOX series data-under-voice, `DUV`.

At the end of a pass, an image is rendered from waterfall data, then it and other demodulated signal data is placed in a queue for later upload to the Network. Logs and other reports are also sent back to the Network by other API actions.

Raspberry Pi 3, Raspbian image

The reference ground station is based on the Raspberry Pi 3 single-board computer.[28] A custom operating system image based on Raspbian is available that includes all necessary software for setting up a ground station client.

Figure 11 shows a screenshot of the `satnogs-config` terminal program for configuring client parameters such as API key and custom pre/post observation scripts. These scripts, if configured, are written by the end user to properly startup and shutdown any additional unique services or configuration, such as antenna switches.



```
                       Basic Menu
[UP], [DOWN] arrow keys to move
[ENTER] to select

        SATNOGS_API_TOKEN          Define API token
        SATNOGS_NETWORK_API_URL    Define network API URL
        SATNOGS_RX_DEVICE          Define RX device
        SATNOGS_STATION_ELEV       Define station elevation
        SATNOGS_STATION_ID         Define station ID
        SATNOGS_STATION_LAT        Define station latitude
        SATNOGS_STATION_LON        Define station longitude
        HAMLIB_UTILS_ROT_ENABLED   Enable Hamlib rotctld
        HAMLIB_UTILS_ROT_OPTS      Define Hamlib rotctld options

                  <  OK  >            <Cancel>
```

Figure 11: `satnogs-config` **on Rasbperry Pi 3**

GROUND STATION

The SatNOGS Ground Station sub-project encompasses the hardware at a specific receiving location including computer, rotator, antennas, and RF path. Figure 12 illustrates the modular nature of the system.

Computer

Ground station owners execute the Client software on either an embedded single-board computer such as a Raspberry Pi 3 or on a PC. This computer also, via network TCP connections, communicates with the rotator controller and receiver hardware. A software-defined radio receiver, such as the inexpensive RTL2832U-based USB device,[29] is attached to the computer and the connection configuration setup with the `satnogs-config` utility.

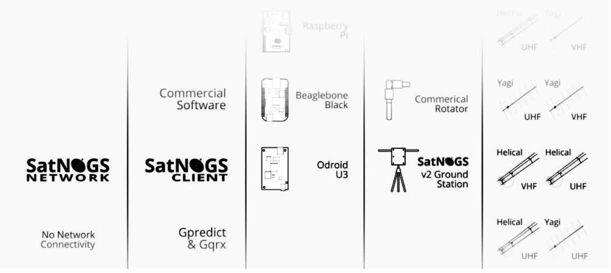


Figure 12: Ground Station modular options.

Rotator and Antennas

The rotator and antennas may be any combination of commercial or custom hardware. Figure 13 shows a commercial rotator with a pair of commercial antennas. Omnidirectional antennas, such as in Figure 14 require the minimum of hardware for a functional SatNOGS ground station.

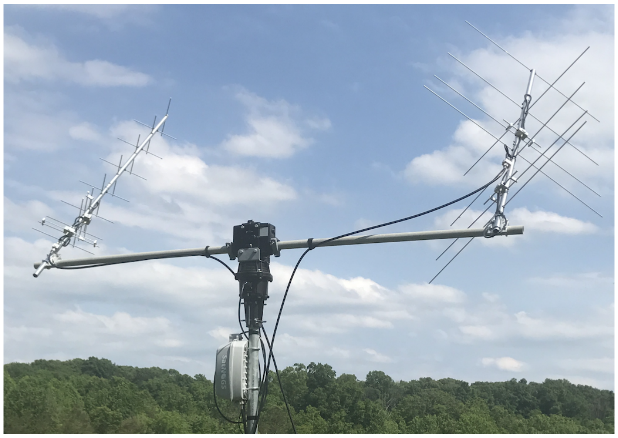


Figure 13: Corey Shields KB9JHU ground station #2 with commercial rotator with UHF and VHF antennas.

Designs and instructions are available within the project for a 3D-printed rotator, shown in Figure 15.

Because the Ground Station rotator's controller implements the common EasyComm 2 protocol, it acts like any other homebrew or commercial rotator controller using the same format. The Client software therefore works exactly the same with the SatNOGS rotator design or with an existing station's rotator.

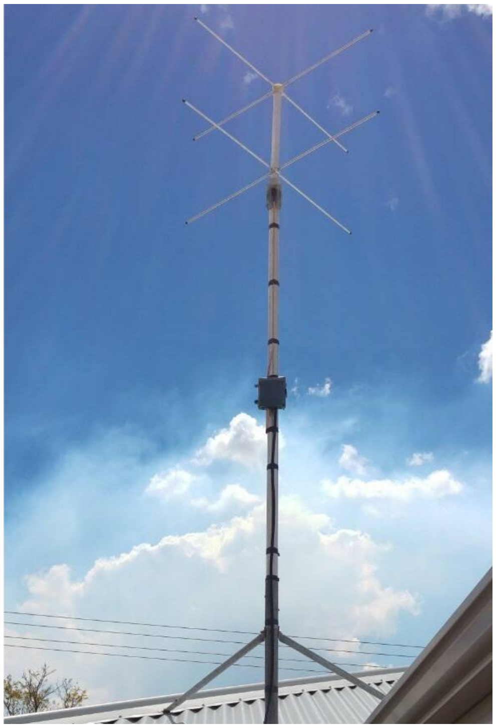


Figure 14: Omnidirectional antenna for Rick Zathras' ground station #23 in Perth, Australia.

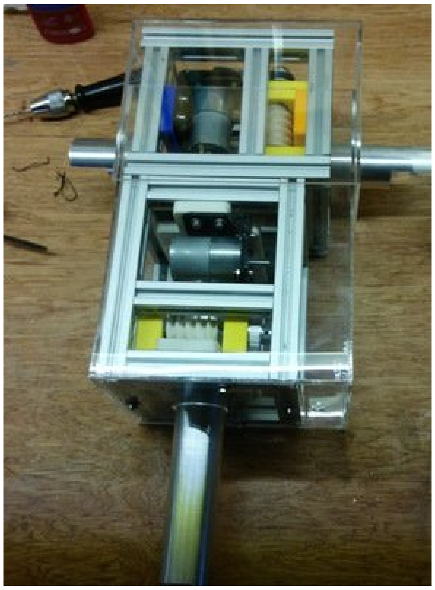


Figure 15: Version 3.0 of the SatNOGS Rotator design.

There are designs for every component necessary for a SatNOGS-compatible ground station as part of the project. Antennas and RF path hardware may be use SatNOGS, homebrew, or commercial as the station owner deems appropriate. To help the goal of easily constructed ground stations, the project's designs focus on common hardware, 3D-printed parts, and tools available in most experimenters' shops of local "maker spaces."

CONCLUSION

The SatNOGS project aims to provide and promote free and open source satellite ground stations. Modern open software and web technologies are used to coordinate these stations to more fully utilize the reception capabilities for low earth orbiting satellites. By using a modular approach to the ground station segment, the existing stations of radio amateurs and others may be used with the network. Avoiding custom, network-specific software and hardware and ensuring all design information, code, and received data is and remains freely available is a core tenet of the project. Individuals and organizations are encouraged to partner with the project to help realize these goals.

REFERENCES

[1] SETI@home. U.C. Berkeley. http://setiathome.berkeley.edu/

[2] European Space Agency. Global Educational Network for Satellite Operations. http://www.esa.int/Education/Global_Educational_Network_for_Satellite_Operations

[3] RBC Signals. http://rbcsignals.com/

[4] Kongsberg Satellite Services (KSAT). https://www.ksat.no/

[5] The FUNcube Project. http://funcube.org.uk/

[6] Hackerspace.gr. Athens, Greece. https://www.hackerspace.gr/

[7] https://2014.spaceappschallenge.org/project/satnogs/

[8] Hackaday. The 2014 Hackaday Prize. https://hackaday.io/prize/2014

[9] Libre Space Foundation. http://docs.satnogs.org/

[10] UPSat. https://upsat.gr/

[11] HamRadioNow. Adventures of a Hacker Turned Ham, Michael Ossmann. TAPR/AMSAT Ban-

quet, Dayton 2015. https://www.youtube.com/watch?v=LpSlGKqeZ4I

[12] W. Silver N0AX. A Maker's Introduction to Ham Radio. Makezine. http://makezine.com/2015/06/30/a-makers-introduction-to-ham-radio/

[13] SatNOGS | Satellite Networked Open Ground Station. http://satnogs.org

[14] SatNOGS Code and Documentation Repository | GitLab. https://gitlab.com/librespacefoundation/satnogs

[15] SatNOGS Documentation. https://docs.satnogs.org/

[16] SatNOGS WiKi. https://wiki.satnogs.org

[17] SatNOGS Network. https://network.satnogs.org

[18] SatNOGS Network (Staging). https://network-dev.satnogs.org

[19] ARISS - Amateur Radio on the International Space Station. http://www.ariss.org/

[20] Internet Archive collection of audio recordings from observations from the SatNOGS network. https://archive.org/details/satnogs

[21] AMSAT-NA AO-85 (Fox-1A) CubeSat. https://www.amsat.org/two-way-satellites/ao-85-fox-1a/

[22] ISS Detector | The mobile satellite tracker. http://www.issdetector.com/

[23] T. S. Kelso. CelesTrak. http://celestrak.com/

[24] D. Estevez. gr-satellites | GNU Radio decoders for satellite. https://github.com/daniestevez/gr-satellites

[25] PyEphem. http://rhodesmill.org/pyephem/

[26] hamlib | Ham Radio Control Libraries. http://sourceforge.net/projects/hamlib/

[27] GNU Radio. http://gnuradio.org/

[28] Raspberry Pi. https://www.raspberrypi.org/

[29] Osmocom. rtl-sdr. http://sdr.osmocom.org/trac/wiki/rtl-sdr