

Valparaiso University ValpoScholar

Engineering Faculty Publications

College of Engineering

10-2015

SatNOGS: Satellite Networked Open Ground Station

Daniel J. White

Valparaiso University, dan.white@valpo.edu

Ioannis Giannelos

Libre Space Foundation

Agisilaos Zissimatos

Libre Space Foundation

Eleytherios Kosmas

Libre Space Foundation

Dimitrios Papadeas

Libre Space Foundation

See next page for additional authors

Follow this and additional works at: http://scholar.valpo.edu/engineering_fac_pub

 Part of the [Engineering Commons](#)

Recommended Citation

White, Daniel J.; Giannelos, Ioannis; Zissimatos, Agisilaos; Kosmas, Eleytherios; Papadeas, Dimitrios; Papadeas, Pierros; Papamathaiou, Matthaios; Roussos, Nikolaos; Tsiligiannis, Vasileios; and Charitopoulos, Ioannis, "SatNOGS: Satellite Networked Open Ground Station" (2015). *Engineering Faculty Publications*. Paper 40.
http://scholar.valpo.edu/engineering_fac_pub/40

This Conference Proceeding is brought to you for free and open access by the College of Engineering at ValpoScholar. It has been accepted for inclusion in Engineering Faculty Publications by an authorized administrator of ValpoScholar. For more information, please contact a ValpoScholar staff member at scholar@valpo.edu.

Authors

Daniel J. White, Ioannis Giannelos, Agisilaos Zissimatos, Eleytherios Kosmas, Dimitrios Papadeas, Pierros Papadeas, Matthaios Papamathaiou, Nikolaos Roussos, Vasileios Tsiligiannis, and Ioannis Charitopoulos

SatNOGS: Satellite Networked Open Ground Station

Daniel J. White, Ph.D., AD0CQ

Valparaiso University
Valparaiso, Indiana
dan.white@valpo.edu

Ioannis Giannelos, Agisilaos Zissimatos, Eleytherios Kosmas,
Dimitrios Papadeas, Pierros Papadeas, Matthaïos Papamathaiou,
Nikolaos Roussos, Vasileios Tsiligiannis, Ioannis Charitopoulos

Libre Space Foundation
Athens, Greece
info@satnogs.org

Abstract—The SatNOGS, or Satellite Network Open Ground Stations, project promotes and supports free and open space applications. It seeks to solve the problem of connecting many satellite users/observers to many ground station operators. Modern open 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.

Index Terms—SatNOGS, CubeSat, software-defined radio, satellite ground station, open source

I. INTRODUCTION

The SatNOGS [1] project seeks to build a full stack of open technologies for satellite ground stations. Civilian satellite launches have been in a state of change in recent years from the introduction of very small spacecraft which use standardized launch carriers such as the CubeSat and PPOD specifications [2]. This has lowered the bar to satellite ownership and their availability for educational and amateur projects and citizen science.

Figure 1 shows that the number of CubeSat-class satellite launches has increased nearly exponentially since the first in 2000. Previously the domain

of University projects, the last 3 years have seen a huge increase in non-government or university launches. These civilian satellites include commercial, like Planet Labs’ Flock-1 satellites [3], non-profit, like The Planetary Society’s recent LightSail [4], and amateur, like AMSAT-NA’s upcoming Fox-1 series [5].

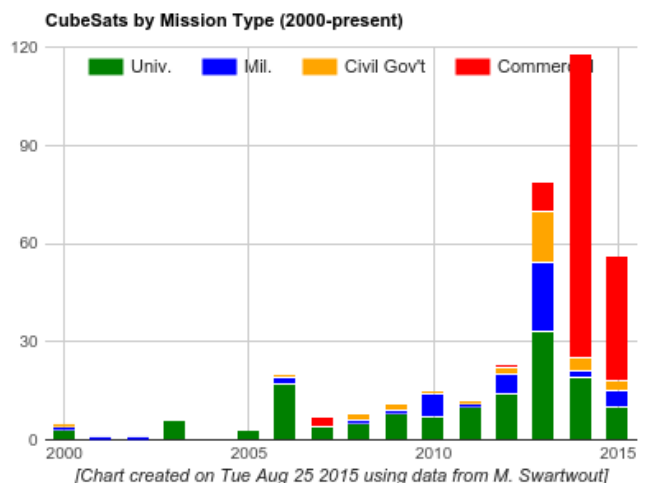


Fig. 1. CubeSat launches per year through 2015-07-17, from [6]. The “Commercial” category includes non-profit and amateur satellites.

Each satellite owner typically operates their own ground station for command and control. The low-

earth orbits (LEO) of these spacecraft result in short time windows when the spacecraft is above the local horizon for communication. As a result, owners seek to enlist the help of other suitably equipped stations for collection of data. The FUNcube project is a prime example of a well-organized effort to receive and collect data from satellite for educational outreach [7].

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.

Once an individual or organization builds a ground station, 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 [8].

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) [9] 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.

Members of *Hackerspace.gr* [10] 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 [11]. It was later submitted as an entry to the 2014 Hackaday Prize, ultimately winning the grand prize of \$196,418 [12]. The funds provided seed money to start the Libre Space Foundation [13], a new non-profit dedicated to supporting free and open source space and related projects.

A timely and unique aspect of the project is its bridging of the Maker / Hacker and the Amateur Radio communities. The 2015 TAPR/AMSAT Banquet's speaker, Michael Ossmann AD0NR pointed out the many characteristics valued and shared by these two communities [14]. Ward Silver, N0AX's article for Makezine also does a good job of drawing these connections [15].

This paper seeks to give an overview of the SatNOGS project's major components. Figure 2 gives a high-level overview of the relationship between users and ground stations. Figure 3 and the following sections describe the four major sub-projects: Network, DB, Client, and Ground Station. The modular approach maximizes use of already available components at a ground station.

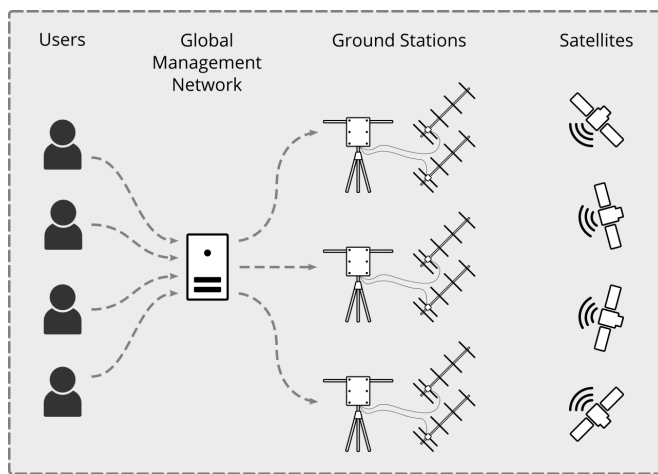


Fig. 2. Overall view of the SatNOGS concept.

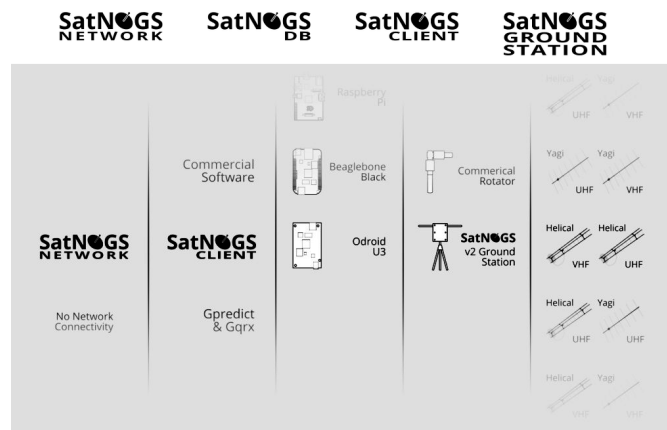


Fig. 3. The four sub-projects are designed with a modular approach with well-defined interfaces, allowing the ability to relatively easily interface with existing ground stations.

All code and hardware designs for the project are publicly available under Open Software (AG-PLv3 and GPLv3) and Open Hardware (CERN OHLv1.2) licenses [16]. Software documentation may be found at [17]. Note, screen captures of the various software components, hardware pictures, and demonstrations are included in the more appropriate format of the presentation instead of in the paper.

II. 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 4 shows this process as a diagram.

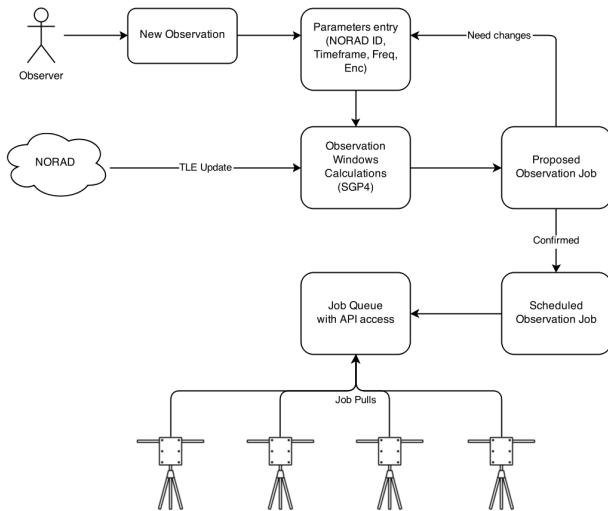


Fig. 4. Diagram of a user scheduling an observation on the network.

Ground Stations collect observation data then send it back to the Network. 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.

Calculation of candidate times when the target satellite is visible from an active ground station is performed with the assistance of the PyEphem library [18]. The library accepts orbital elements for the satellite, Ground Station locations, and the desired time frame. With higher densities of Ground Stations which can see multiple satellites, this scheduling can be optimized for many factors.

Because all code and documentation of all parts of the project are free and publicly available, anyone is able to contribute to these improvements. Indeed, this open collaboration is one of the SatNOGS project’s founding principles.

III. 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 the same spirit of the rest of the project, the database is open access and not specific only to the SatNOGS project.

Specifically, DB is a crowd-sourced suggestions app for transponder data. Satellites are identified by their NORAD (now USSPACECOM) 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.

Updates are accepted from the public and 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 capable browser.

IV. CLIENT

SatNOGS Client consists of software running a computer which controls the ground station hardware. Figure 5 diagrams the internal (modular!)

components of the client’s interaction with the Network.

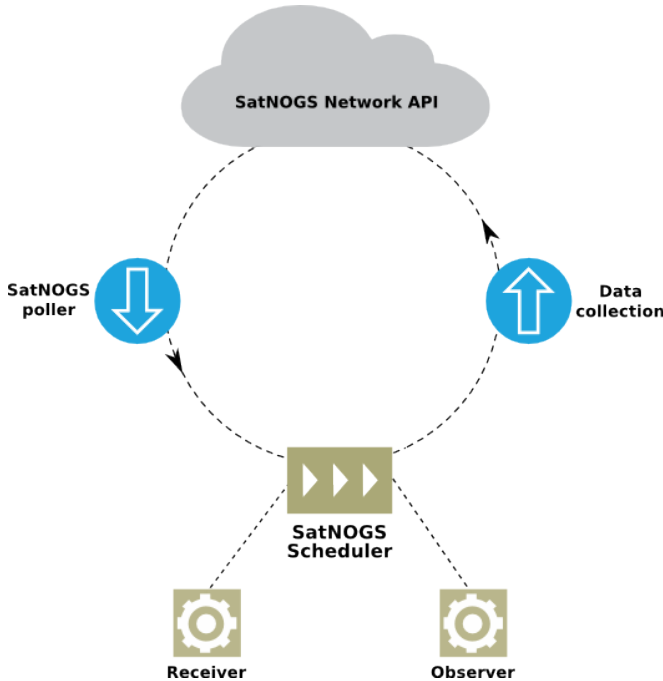


Fig. 5. Client software components and interactions.

The poller regularly checks the Network API for observation jobs scheduled for the local Ground Station. Information contained in each job includes satellite orbital elements, receiver tuning and demodulation parameters, and timing. The PyEphem [18] 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 [19]. 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.

External programs for starting the receiver driver are spawned before a scheduled observation and setup to listen for `rotctl` commands for doppler frequency corrections throughout a pass. For use with the popular Realtek RTL2832U DVB-T dongles, the SatNOGS group maintains a version of `rtl-sdr` [20] software’s `rtl_fm` utility which accepts frequency control via `rigctl` commands

over a network port [21]. Code for interfacing the Client with other SDR software is in progress, including receivers using GNU Radio [22]. Radios whose baseband signals enter the computer via soundcard are also planned.

At the end of a pass, 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.

V. GROUND STATION

The SatNOGS Ground Station sub-project encompasses the antennas, rotator, and RF path. Ground station owners execute the Client software and build or configure the hardware. Plans and instructions are available for the v2 ground station and are being finalized for the updated v3 version. Gears and other parts are 3D printed and the rest of the hardware for the ground station is easily available. Besides access to a 3D printer, only hand tools are required for building either version.

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.

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.”

VI. 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] SatNOGS — Satellite Networked Open Ground Station. <http://satnogs.org>
- [2] CubeSat — Wikipedia. <https://en.wikipedia.org/wiki/CubeSat>
- [3] Planet Labs. <https://www.planet.com/>
- [4] The Planetary Society. <http://planetary.org/>
- [5] AMSAT North America — The Radio Amateur Satellite Corporation. <http://www.amsat.org/>
- [6] M. Swartwout. CubeSat Database. <https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database>
- [7] The FUNcube Project. <http://funcube.org.uk/>
- [8] SETI@home. U.C. Berkeley. <http://setiathome.berkeley.edu/>
- [9] European Space Agency. Global Educational Network for Satellite Operations. http://www.esa.int/Education/Global_Educational_Network_for_Satellite_Operations
- [10] Hackerspace.gr. Athens, Greece. <https://www.hackerspace.gr/>
- [11] <https://2014.spaceappschallenge.org/project/satnogs/>
- [12] Hackaday. The 2014 Hackaday Prize. <https://hackaday.io/prize/2014>
- [13] Libre Space Foundation. <http://docs.satnogs.org/>
- [14] HamRadioNow. Adventures of a Hacker Turned Ham, Michael Ossmann. TAPR/AMSAT Banquet, Dayton 2015. <https://www.youtube.com/watch?v=LpSIGKqeZ4I>
- [15] W. Silver N0AX. A Maker's Introduction to Ham Radio. Makezine. <http://makezine.com/2015/06/30/a-makers-introduction-to-ham-radio/>
- [16] SatNOGS Code and Documentation Repository — GitHub. <https://github.com/satnogs>
- [17] SatNOGS Documentation. <http://docs.satnogs.org/>
- [18] PyEphem. <http://rhodesmill.org/pyephem/>
- [19] hamlib — Ham Radio Control Libraries. <http://sourceforge.net/projects/hamlib/>
- [20] Osmocom. rtl-sdr. <http://sdr.osmocom.org/trac/wiki/rtl-sdr>
- [21] SatNOGS. rtl-sdr — rtl_fm with rigctl. <https://github.com/satnogs/rtl-sdr>
- [22] GNU Radio. <http://gnuradio.org/>