18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.
Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

SpaceOps-2025, ID # 543

An Overview of the Ground Automation of Operational Procedures for EIRSAT-1, a 2U CubeSat Project

Joseph Fisher[a,*], David Murphy[a,b], Joseph Thompson[b,c], Caimin McKenna[a,b], Gabriel Finneran[a], Laura Cotter[a], Cúan de Barra[a], Aaron Empey[a], Fionn Gibson Kiely[c], Padraig McDermott[a], Sheila McBreen[a,b], David McKeown[b,c], Lorraine Hanlon[a,b], Maeve Doyle[a], Rachel Dunwoody[a], Jack Reilly[a], Ronan Wall[a,b], Antonio Martin-Carrillo[a,b]

[a] *School of Physics, University College Dublin, Ireland*
[b] *UCD Centre for Space Research, University College Dublin, Ireland*
[c] *School of Mechanical and Material Sciences, University College Dublin, Ireland*
* Corresponding Author: *joe.fisher@ucdconnect.ie*

Abstract

EIRSAT-1 is a 2U CubeSat which was designed, tested, and built in University College Dublin (UCD). It was launched in December 2023 to a Sun-synchronous orbit of an altitude of ~520 km. It carries experiments for high-energy astrophysics and materials science, along with an attitude control testbed, all of which were developed in UCD. The mission's operations team consists of ten postgraduate and post-doctoral researchers, each of whom spends a portion of their time on operations, in addition to their own research. Prior to launch, operators underwent extensive training using realistic simulations of the expected short communication windows, incorporating the bandwidth limitations imposed by the use of VHF/UHF. As a result of this training, and the team's growing experience with in-orbit operations, an array of tools have now developed to streamline operational procedures and provide automated on-ground data handling, storage, and visualisation. Passive tools, such as automatically generated contact-planning documentation, and in-pass tools, such as Python scripts and a custom-built Python library used to communicate with the spacecraft, are in development. These are tested in a sandbox environment utilising a 'FlatSat' configuration before in-orbit testing and deployment. The suite of tools developed to date has made the operations of EIRSAT-1 manageable with a small team, while ensuring the spacecraft's health is well monitored, and essential science and house-keeping data are downlinked efficiently. The focus of the team's ongoing work is the implementation of more tools to further reduce operator workload and progress automation of spacecraft operations.

Keywords: CubeSat, Automated Operations, Operations Development

Acronyms/Abbreviations

Gamma-ray Burst (GRB) / Gamma-ray Bursts (GRBs)
Housekeeping Data (HK)
Launch and Early Operations Phase (LEOP)
Mission Control Software (MCS)
Mission Test (MT)
On-board Software (OBSW)
PostgreSQL database (PSQL)
South Atlantic Anomaly (SAA)
Telecommands (TCs)
Telemetry (TM)
Terminal Node Connector (TNC)
University College Dublin (UCD)

1. Introduction

1.1 Mission Overview

EIRSAT-1 (Figure 1) is a 2U CubeSat designed, tested and built in University College Dublin (UCD) by a student-led team with support from ESA *Fly Your Satellite!* program. A detailed description of the spacecraft and its subsystems can be found in Murphy et al. (2018). After a successful Launch and Early Operations Phase (LEOP), the spacecraft is operating nominally, and experiments are being optimised for in-orbit operations. There are three on-board experiments; GMOD (Murphy et al., 2022, 2023), for detecting gamma radiation from astrophysical gamma-ray

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

bursts (GRBs); EMOD, a material science experiment, and WBC, an attitude control and determination software test bed.

EIRSAT-1 was launched on December 1st 2023 and was placed into a Sun-synchronous orbit with an altitude of ~520 km, corresponding to an orbital period of ~95 minutes. Two ground stations are used to communicate with the satellite: one in UCD's Belfield Campus (BFD), the other in Dunquin / Dún Chaoin, south-west Ireland (DQN), approximately 310 km apart. BFD has capabilities for both sending telecommands (TCs) and receiving telemetry (TM), while DQN currently only receives TM[1]. In this orbit, there are two communication windows per day during which the spacecraft is visible from our primary ground station, BFD, and two-way communication is possible, one in the morning (~9 am - 1 pm UTC) and the other in the evening (~8 pm - 12 am UTC). In each of these windows, a maximum of three contacts above 5º elevation for BFD, can occur, each lasting ~8-10 minutes. This allows a maximum of approximately one hour of two-way communication time each day. This time constraint is clearly not optimal for operating three payloads, while also downlinking vital data to monitor the health of the spacecraft continuously. To counter this, automated contact tools are being developed to decrease operator overhead during contacts, reduce the time needed to complete procedures, and thereby allowing greater operational efficiency during the limited pass windows.

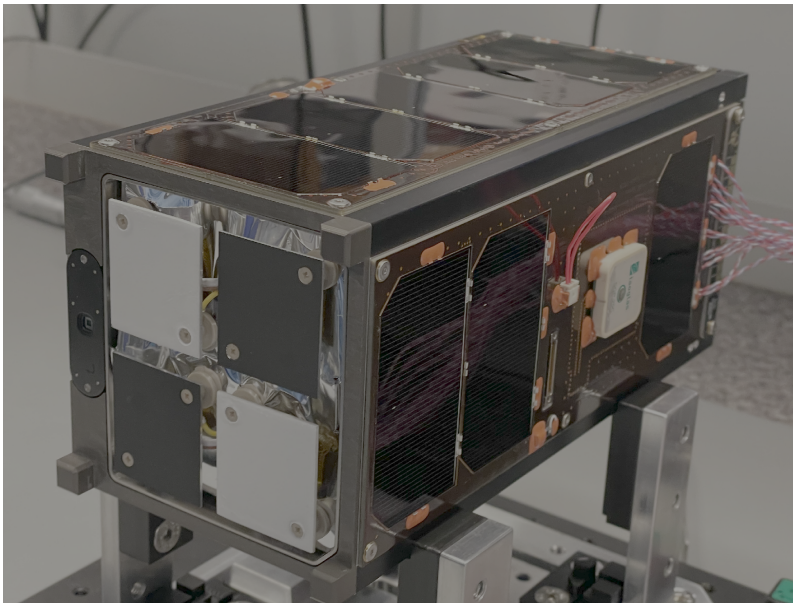

Figure 1: The EIRSAT-1 Flight Model

1.2 Operations and the Operations Team

The EIRSAT-1 operations team consists of ten postgraduate and postdoctoral researchers. Each of the communication windows described in §1.1 constitutes a 'shift' for which at least one operator is required to communicate with the satellite. During a shift, the operator must perform a predetermined set of tasks, including downlinking data to verify the health of the spacecraft, downlinking scientific/experimental data from the payloads, and executing procedures to initialise, change settings, or turn off payloads, depending on the goals and priorities for the pass. Completing these tasks has proven to be very challenging given the short time available for each pass.

During LEOP, two operators were needed for each shift, requiring 28 shifts (7 days x 2 windows x 2 operators) to be completed each week. This led to operators working 2-3 times per week, during both nights and weekends. These hours caused burnout for operators, and negatively affected operators' research external to the mission. To counter this, in the months following LEOP, shifts during evening and weekends were restricted to only downlinks and hence only one operator was required to complete this, totalling 17 shifts per week. Since automated operations have commenced, only one operator is now required to monitor contacts, reducing the workload to 11 shifts per week - 2 operators each day Monday, Tuesday, Thursday and Friday, and only 1 operator for full days on Wednesday, Saturday and Sunday.

[1] At time of submission, DQN has all necessary hardware for sending TCs, and testing is currently underway.

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

1.3 Pre-launch Operator Training

Prior to launch, multiple Mission Test (MT) campaigns were carried out during which the on-board software (OBSW) was tested and verified to be in a flight-ready condition (Doyle, et al. 2022). During these campaigns, new members of the team were also trained for operations by the senior members of the team, and the operations manual developed for in-orbit operations was validated (Dunwoody, et al. 2023). The longest of these tests lasted three weeks, with operators completing simulated passes with realistic VHF/UHF limitations. Operators interfaced with the spacecraft using Mission Control Software (MCS) from Bright Ascension[2], which is still used for operations. All contact planning documentation (including a 'Daily plans' document, a document containing objectives and results for each contact, and a close-out document) were generated manually by the operators using Google Cloud services, creating large overhead during the operations. Any data downlinked during the contacts had to be renamed and stored appropriately for accessibility, as well as uploaded to Grafana[3], an open-source data visualisation software, for analysis. It was realised that these repetitive tasks, while easy to do, could be automated to allow more time to be spent analysing downlinked data and planning for future contacts based on these data. These tools are discussed further in §3.

1.4 In-orbit Operations

During LEOP, the spacecraft was fully configured for in-orbit operations after a successful launch and deployment. Because the operational procedures during this period would only be performed once (for initialising subsystems and payloads, for example), no automated tools for these procedures were developed. However, as the team moved towards nominal operations, it was noticed that many tasks were repeated in each pass or in each shift. Such tasks include connecting to the ground station to enable two-way communications, initiating a contact with the spacecraft and commencing the sending of live events from the spacecraft during contacts, downlinking essential data, such as housekeeping data (HK), and uplinking essential payload commands. These repetitive tasks motivated the development of automated tools for in-pass operations. Due to the reduced operator overhead associated with automation, more procedures can be completed and more data downlinked. As well as this, shifts can now be carried out fully autonomously without the need for a human operator. Instead, operators can be 'on call' if an error is raised by the scripts and fix any problems that may arise. This reduces the number of shifts which need to be taken by operators, hence reducing the workload on the small operations team.

2. Ground Architecture

2.1 Server

A dedicated in-house Linux server is used for operations and data storage. All of the required tools run on this server, including data visualisation, as well as all of the automated tools. A local network is used to connect the in-house server with the operations consoles and the BFD ground station. To connect to the DQN GS, a port on the server is mapped to a port on the DQN GS via an SSH tunnel (See Figure 2). The direct local connection from the operations consoles to the BFD GS allows any passes to be completed manually in case of server fault. The server has a second direct internet connection in case of local network failure. While in this scenario operations would not be possible, DQN can still be controlled and any beacons decoded without the need of MCS.

Raw data downlinked from the spacecraft is stored as .dat files in a dropbox on the server. Using Samba[4], this dropbox is made accessible on the Windows Operating Consoles, allowing data downlinked during manual passes to be easily uploaded, backed up and accessed in Grafana.

Docker[5] is used to containerise the tools on the server, allowing easy single-tool management, such as making changes and updates without affecting other tools. In case of any server failures, such as power cuts, these docker containers are set to restart themselves, so no loss of services is experienced.

All scheduled tools are handled by Rundeck[6], an open-source software, which runs in a docker container on the server. Rundeck utilises 'jobs' which can run a series of scripts in a desired order. Jobs can be reliably scheduled and run without the need for operator intervention and hence allow fully autonomous operations. These jobs are sorted by project depending on their use and the resources which they can access while running. For the EIRSAT-1 mission, it was found that having three projects - one for ground station control, one for MCS control, one for repeated non-

[2] https://brightascension.com/products/mission-control-software/
[3] https://grafana.com/
[4] https://www.samba.org/
[5] https://www.docker.com/
[6] https://www.rundeck.com/

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

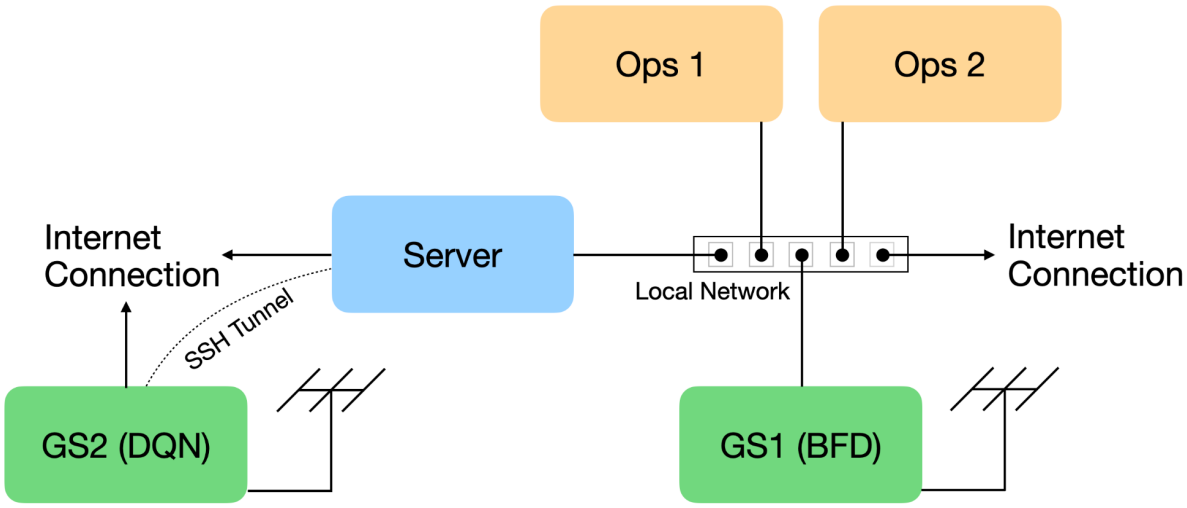

Figure 2: The ground segment architecture used in EIRSAT-1 operations.

contact tasks - best meets the requirements. Each project can connect to and run on different nodes, for example, another docker container on the server or another computer, as required. This organisation reduces the likelihood of jobs executing on incorrect nodes and prevents any errors that may arise. Services which operators may require (Rundeck, Grafana, Remote Connection, etc) are hosted publicly, using Authelia[7] to allow operators and users to log into these services.

Deaggregated data from the spacecraft are stored in an Influx database[8]. This allows easy storage of time series data, which can be displayed in Grafana. For data that does not need to exist in a time series, PostgreSQL[9] (PSQL) is used. Both databases are run in individual docker containers on the server and can be interfaced using Python. This allows data to be read and written with existing tools, rather than inputting data into scripts when required.

2.2 Operations Consoles

Operations are carried out on one of two operations consoles (OPS1 / OPS2) which run Windows 10. The OPS consoles are identically set up, each containing its own MCS instance, and hence either can be used to operate at any time. Two are used only for redundancy in case of failure of either computer. The OPS consoles can access the internet via the internal network configuration. In case of external internet loss, the OPS consoles can still be used to access services hosted on the server, and hence operations can be performed. No rundeck jobs are deployed to the OPS consoles. Instead, rundeck jobs which run on the server can have the ability to connect to MCS on either OPS console.

2.3 Ground Station Computers

Each ground station has its own computer (GS1 in BFD, GS2 in DQN) which controls antenna pointing and radio settings during contacts. Rundeck jobs are deployed on each of the GS computers to handle passes. Both ground stations are near-identical, apart from some minor hardware differences (see §4.2.2). To receive TM from the spacecraft, both groundstations utilise a Pluto SDR, which receives signal in a wide range of frequencies. For sending TCs, ICOM radios are used. Both of these connect to the downlink and uplink cross-yagi antennas respectively. Rundeck jobs which are deployed onto the ground station computers are used to automatically control both the antenna pointing and the doppler offset of sending and receiving, which both utilise two in-house custom built python packages. These are discussed further in §4.2. When the ground stations are not being used for operations, they can be controlled and used for SatNOGs observations, the software for which runs in a docker container on each of the GS computers.

[7] https://www.authelia.com/
[8] https://www.influxdata.com/
[9] https://www.postgresql.org/

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

3. Passive Tools

Many tools used in day-to-day operations do not depend on contact times and are run periodically. These tools do not require any direct connection to the spacecraft, but complete tasks that will aid the operations team. As discussed in §2.1, these tools are controlled by Rundeck. The scripts are executed on another docker image with Python, and that cannot connect to the OPS or GS computers. This avoids any potential problems with tools running during a contact, which may cause the docker image and tools to crash.

3.1 Contact Documentation

The first automated tool was the generation of contact planning documents. These documents are more detailed than those used during pre-launch training, and now contain; (1) Information of upcoming contacts, including AOS time, LOS time and peak elevation for BFD, from where TCs are sent; (2) A pre-contact checklist to ensure that both BFD and DQN are in a state to carry out the contact; (3) A table to populate with last known spacecraft state, so that any deviations (ie a mode change) can be easily noticed; (4) Space to plan out the procedures from the operations manual to be completed in the pass, and the steps at which to begin if they are continued from the last contact; (5) Space to make notes on the downlinked data; (6) A table to populate with the last known spacecraft state, which is passed forward to the next contact document. This information is essential in a team setting where multiple operators may be operating in the same day, and procedures may need to be resumed between passes.

Using Google Cloud services through its Python API[10], these documents are now created automatically for each upcoming contact, according to the upcoming contact times. This allows operators more time to familiarise themselves with the current spacecraft state and of any procedures that need to be completed. Google Cloud services are also used to create a calendar of contact times which all operators could easily check and confirm details of the upcoming shift, including contact AOS and LOS times, as well as the peak elevation of each contact.

The job which creates contact documentation runs in conjunction with jobs which schedule contact jobs (discussed further in §3.5). In this procedure, the TLE is automatically fetched from Celestrak[11] each hour and saved onto the server so that it is accessible to all other jobs. After this procedure, the contact documentation is created based on the contact times using the TLE. The *skyfield* Python package is used to find pass times for each ground station and hence contact times where the satellite has overlapping passes. If a document already exists with the correct contact information, a duplicate is not created.

3.2 GMOD Commands and GRB data

Due to the Sun-synchronous orbit of EIRSAT-1, it passes through both polar regions and the South Atlantic Anomaly (SAA). These regions of high radiation cause sharp increases in the gamma-ray counts observed by GMOD, which can cause the instrument to reset due to an excessive count rate. However, the OBSW has the ability to execute time-tagged commands, referred to within the OBSW as 'TimeActions'. With this capability, a list of times can be generated when GMOD should switch between 'idle' and 'experiment' mode. Using AP8/AE8 models of trapped particles (Sawyer and Vette, 1977; Vette 1991), of Earth's radiation belts, approximate polygon models of regions where GMOD must switch to idle mode are identified. With the most up-to-date TLE, times at which GMOD must switch from experiment to idle mode (i.e. entering a radiation region) and vice versa (i.e. leaving a radiation region) are found. The times are formatted correctly for uplinking to the spacecraft, and a file is generated. This automated procedure runs once per day and uploads these files into the applicable contact folder on Google Drive. These are manually uplinked to the spacecraft each shift, and command GMOD autonomously for ~12 hours.

After GMOD gamma-ray time series data has been downlinked, it is important to identify events of possible astrophysical origin. General Coordinates Network (GCN) data, containing event times for GRBs from other high energy satellite missions, such as NASA's Swift and Fermi, or CAS' Einstein Probe, are automatically fetched each hour and marked on the downlinked data. This feature makes spotting a potential GRB in the downlinked time series much easier and avoids false alarms.

3.3 Operator Schedule

The operations team follows a schedule of ~11 shifts per week (see §2.1). This rota is made manually by the team monthly and saved to Google Cloud Services as a .csv file. This file is read automatically every day by a rundeck job and saved as a second Google calendar for operators who are on shift for the coming week. Four days a week, there

[10] https://github.com/googleapis/google-api-python-client
[11] https://celestrak.org/

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

are operators on both the morning and evening shifts, while on one working day each week and on the weekends, there is only one operator for the whole day. During these full days the operator only completes one shift of passes, however, they must be on call to handle any anomalies encountered by the contact tools during the shift that is not operated. For this reason, shifts are broken down as follows:

- Morning shift: 8 am - 4 pm UTC
- Evening shift: 4 pm - 12 am UTC
- Full day shift: 8 am - 12 am UTC

When an operator is on call, the Grafana OnCall alert system is used to alert operators of any anomalies encountered. This is discussed further in §5.

3.4 TMTC Database

In the first days of the mission, beacons observed by other ground stations around the world proved invaluable to the team, allowing an accurate picture of the health of the spacecraft. These beacons were manually pulled from SatNOGs[12], an open-source network for ground stations observing satellites with public observations. Audio files from these observations were manually downloaded and passed through an audio decoder to extract the data from beacons. This process is now fully automated on the server for beacons and TM received from the spacecraft. These data are pulled from both the EIRSAT-1 ground stations and SatNOGs ground stations. The process to extract data from the received TM is broken down into several phases (see Figure 3):

1. The first stage in this process is to import any data, including parsing audio from SatNOGs observations. Any contacts of EIRSAT-1 that have been made in the past day from SatNOGs are recorded using the *requests* Python package. These are then cross checked with a PSQL table of observations which have already been parsed, and any new observations are returned. Each audio file from the observations is downloaded and parsed, returning byte objects of the beacon which are read for individual frames. These parsed frames are saved into an PSQL table, and the observation is marked as processed. This process is repeated continuously, and sleeps for three minutes before fetching new data. The decoded data from our groundstations is also directly saved as frames in this PSQL table.
2. The next stage is to remove any possible duplicate frames. This issue arises if a SatNOGS observation has taken place at the same time as a commanded contact, and the same data could be decoded twice. Only unique frames are passed into the next PSQL table.
3. The third stage of the process is to remove any framing from the packets received. First, the frames are parsed, and the headers are read using the *construct* Python library. From this, the start position of each packet can be found, and hence the data can be easily read. For packets that are completely contained in one frame, this procedure is straight forward. However, packets can be split across two frames, where no headers are applied to the second half of the packet. From the header at the start of a split packet, it can be easily determined if a packet has been split by comparing the size of the packet and the size remaining in the frame - if the packet is longer than the remaining space, the packet has been split. Using this, packets which have been split can be reconstructed in their entirety. This process runs continuously, pulling new frames from the unique telemetry frames as they are received live. Any new packets are saved to a telemetry packers PSQL table.
4. Finally, the fourth process reads and decodes all packets in the telemetry packets table. The type of packet can be determined by the packet headers. If the packet is a beacon, its data can be compared to the known beacon structure, and hence parameter values can be obtained from this. If the packet is a response to a 'GET' command, it does not contain the parameter ID, only the ID of TC to which it is replying. This ID can be compared to a PSQL table of the commands sent from the ground stations, and hence the parameter value received can be paired with the parameter ID. These parameter values are stored as an Influx database which can be easily accessed and visualised in Grafana.

One major upside of using this structure for decoding TM is that it increases the chance of decoding data through the use of multiple ground stations with the SatNOGs system. If a full frame is missed by our ground stations, it is possible that an ongoing observation from another ground station caught it, hence the packet is not lost. Moreover, if a packet has been split across two frames as discussed above, it is possible that either one of the frames was not received by the EIRSAT-1 ground station, rendering that (half-)packet unusable. However, if that dropped frame was observed by another ground station, not only are the other packets in the frame recovered, but the packet that was previously unusable can be fully decoded and the data recovered.

[12] https://network.satnogs.org/

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

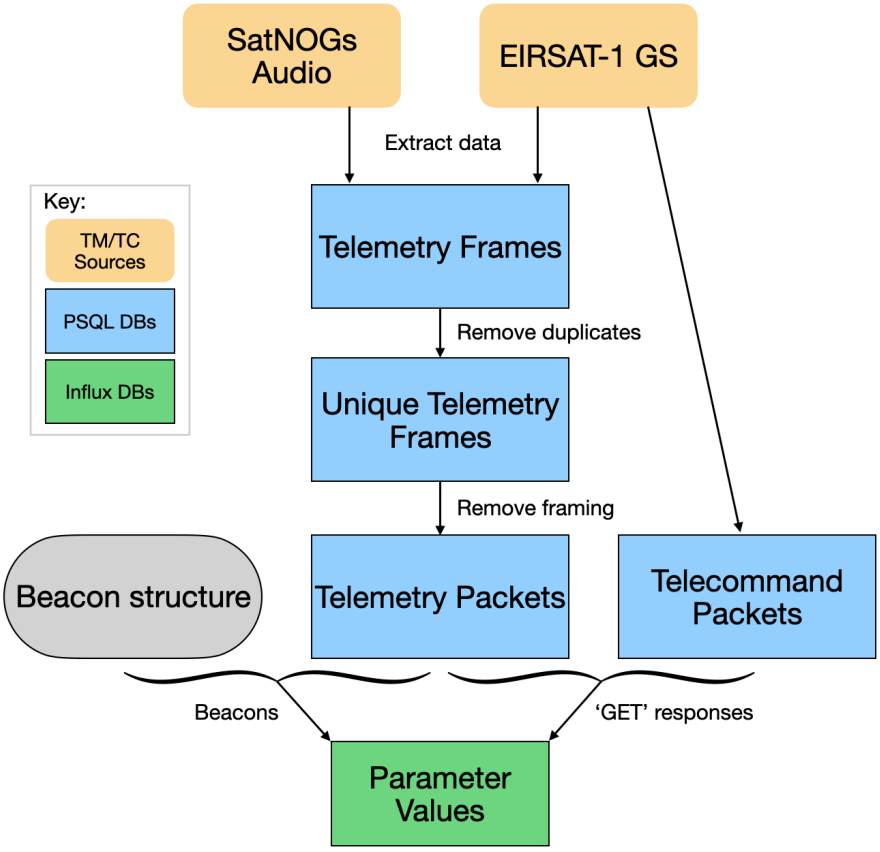

Figure 3: The processes used to decode beacons and parameter 'GET' responses.

3.5 Contact Tool Scheduling

To ensure that tools which require a link to the spacecraft run at the correct times without the need for operators to schedule them, a job which runs hourly is used to schedule these jobs. This is the same job which is used to create the contact documentation. By pulling from the PSQL database containing contact times (mentioned in §3.1), passive tools can be used to schedule contact jobs in Rundeck which run approximately three minutes either side of a contact, allowing all procedures to initialise correctly. Contact tools are discussed further in §4.

4. Contact Tools

Tools that can interface directly with the spacecraft during a contact, along with tools which autonomously control the ground stations, allow fully automated contacts without the need for an operator. They are scheduled to run at times when a link can be made with the spacecraft, as previously discussed. As is the case with passive tools, the contact tools are controlled by Rundeck. Jobs which require a connection to MCS, namely those which can command the spacecraft, are run in a docker image which can connect to the OPS console, while jobs which are designed to control the ground stations (antenna pointing, doppler shifting) are deployed to run locally on the GS computers.

4.1 AutoOps

All of the code and procedures needed for automated spacecraft commanding are contained in a Python package, named AutoOps. Using the MCS Python interface provided by Bright Ascension, a connection from the scripts can be made to MCS on either of the OPS consoles (where OPS1 is the default) and hence can operate the spacecraft. Due to the unpredictable nature of spacecraft operations (for example, a lack of responses to commands, events being raised and transmitted live etc.), the *asyncio* Python package is utilised throughout the AutoOps package. The package consists of two major parts: (i) procedures which can be carried out during the contact and (ii) handlers which can deal with unpredicted / no telemetry from the spacecraft.

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

Before a contact is initiated and automated operations can start, the configuration of the scripts must be correct. This configuration information includes: (i) parameters for setting the connection point to the MCS instance; (ii) information about the spacecraft databases and software images of EIRSAT-1, and (iii) operational configuration, such as times to connect to, and disconnect from, ground stations, waiting periods between sending TCs, and downlink file destinations. These parameters are contained in a file accessible to the Python package and can be changed as required.

4.1.1 Procedures

The initial procedure to be followed in a contact is fixed, regardless of whether the pass is manual or automated. It involves checking the health of the spacecraft before commencing any high-power consumption procedures, such as downlinks or payload commissioning, as well as commencing the sending of live events from the spacecraft to the ground. Early in the mission, this crucial procedure could take several minutes out of the very limited time during a contact because each of the required parameters were requested individually. This procedure was optimised by requesting a HK packet from the spacecraft in the form of a beacon. In the AutoOps package procedures, this beacon is requested and parsed by the Python interface and data for the various parameters are extracted. To confirm that the spacecraft is in a healthy state and ready to proceed with operations, each of these values is checked automatically to ensure it is within its nominal / expected ranges.

After the initial checks are completed, the next procedure is to downlink data from the spacecraft. Data are downlinked from specified channels, each containing data from a unique aggregator. These channels correspond to the on-board Event log, HK, ADCSHK, TED, PASCAL, and various payload data channels (see Appendix A for further channel information). As is the case with manual operations, all new data in the Event log and the previous orbit of HK data are always the first to be downlinked. These data are a priority to understand the present state of the spacecraft, including the mode and software image of the spacecraft, the battery levels, spin rates and payload states, and diagnosis can be carried out using the event log if needed. The remaining data are downlinked in the following order: (1) All new scientific payload data generated; (2) All new spacecraft data (HK, TED, PASCAL and ADCSHK) in the last day; (3) All new spacecraft data in the last week; (4) All new HK and TED in the last month. After this, older data get written over due to the circular nature of the channels. This priority list allows all new data to be downlinked before moving to older data, hence allowing any current problems to be diagnosed, while also ensuring all payload data is downlinked.

Other procedures are currently being developed to further progress automation. These include the ability to uplink a new software image to the spacecraft, as well as autonomously uplinking GMOD TimeActions to the spacecraft.

4.1.2 Handlers

During a contact, many different and possibly unexpected things may occur, such as in-orbit events or issues (live events, lack of responses, unexpected mode change, etc), or ground issues (disconnection from ground stations). These can be managed using asynchronous class functions which wait for certain events to happen or a condition to be met and run when needed.

The most called handler is the beacon handler, which listens to beacons coming in from the spacecraft during a contact. At the beginning of a contact, the beacon is used to initially check the health of the spacecraft before proceeding, as discussed in §4.1.1. As well as this, it is used to ensure the ground HMAC sequence number is in sync with that of the spacecraft. If this is out of sync, any command received by the spacecraft will be ignored. To avoid this, from the beacon received at the start of a contact, the spacecraft sequence number is obtained, and the ground sequence number is updated accordingly. This resyncing behaviour can be toggled at any point in the pass, which is useful when the ground has sent a number of commands with no response. When this happens, with the next decoded beacon the ground sequence number can be resynced. By default, after the first beacon is decoded, this behaviour is disabled to negate the possibility of setting the sequence number to an old one (i.e. when sending TCs and a beacon comes in delayed). In addition to this, when a live event is received during a contact, it is important to inform the operator of its contents and what it could relate to. For this reason, the live event handler only prints out the event received to the log.

Another handler which works parallel to the beacon handler is the spacecraft health check handler. When a beacon is received, this is called to check the live health throughout contacts. The main purpose of this handler is to cancel any procedures if the spacecraft enters safe mode during a pass, hence preventing any high-power consumption in the case safe mode was entered from a low battery voltage. The handler can also print the mode change to the log, alerting operators to the change and commence manual operations.

To deal with the occurrence that the script disconnects from a ground station, a handler is in place to reconnect to it. At the beginning of a contact, the library automatically connects to the ground stations. After this, the connections

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

are periodically checked with a default period of ten seconds. Each ground station is individually checked, and if it is found that one is disconnected, an error is raised to make the operator aware, and reconnection is attempted. Another handler that works parallel to this is the sending handler, which ensures that at least one of the groundstations is marked for sending. At any point in the contact, only one ground station can be used for sending commands to avoid TC interference or duplicate TCs experienced by the spacecraft.

Finally, an LOS handler is in place to deal with the predicted end of contact. This handler waits for the contact to end fully, as marked by last LOS time from the ground stations. When this time is reached, any running procedure is cancelled, and no more procedures are allowed to start. The ground stations are then disconnected from, which avoids any possible errors rising when the ground station pass ends and the MCS connection point is disabled.

4.2 Ground Station Automation

The job which autonomously controls the ground stations is run in three phases: pre-pass, in-pass and post-pass. While this job is controlled by *Rundeck* on the server, it is run locally on each ground station. Because the ground stations are both used as active SatNOGs stations while they are not being used to operate, the pre-pass and post-pass phases are used to stop and restart the SatNOGs software, allowing the ground station hardware to be controlled by our software exclusively. The post-pass phase also backs up any data received to the TMTC Database (§3.4) and backs up the raw downlink waterfalls. The in-pass phase is used solely for controlling the antenna and radio rigs. This is done using a custom-built Python package, *GSctrl,* and *GNURadio*[13], an open-source software used for signal processing (see Figure 4). These only run in times where a contact is expected and are shut off when no EIRSAT-1 contacts are happening, allowing the ground stations to be used by the SatNOGs system. Any attempt to connect to the ground station connection points on MCS while these packages are not running, will raise an error.

4.2.1 GSCtrl

To control the antenna pointing and doppler shifting of uplink and downlink frequencies, *GSCtrl* was developed. Taking the most recent TLE and the coordinates of the groundstation, the azimuths, elevations, and radial velocities of EIRSAT-1 as seen from the ground station are calculated using the *skyfield* Python library. Using the known fixed frequencies for uplink and downlink, the doppler shift throughout the pass is calculated from the radial velocity. *rigctld*[14] and *rotctld*[15] are two daemon programs which can be used to control the frequency settings and the pointing of the antennas. The antennas can be set to track the satellite across the sky and the radio set to the correct shifted uplink frequency by continuously calling these programs with updated parameters.

4.2.2 GNURadio

As well as controlling the antennas and radio frequencies, TM received from the spacecraft must be collected, and TCs must be correctly formed for sending. To do this, *GNURadio* is used to build a flowchart for signal processing, along with custom-built Python flowchart blocks, and blocks from *gr-satellites*[16]. As discussed in §2.3, a PlutoSDR is used to receive TM with a wide bandwidth. Using the doppler shifted frequency calculated with *GSCtrl*, the data in the correct window in which TM is being received can be decoded using *gr-satellites*. The window search allows the signal to remain centred through the decoding process, and to reduce the impact of noise outside of the window. These data are then passed to MCS through the connection point server for use in operations. For sending TCs, commands from MCS are sent through the connection point. Ax.25 headers are then applied and the HMAC hash calculated from the secret key on the GS computers are applied. This is all completed using custom-built Python blocks. These data are audio encoding for sending to the spacecraft. In BFD, this audio encoding is done using a physical Terminal Node Connecter (TNC), while this is completed by software in DQN. These encoded data are sent to the ICOM radio for uplinking, which is set to the correct frequency by *GSCtrl.*

5. Operator Alerts

As previously mentioned in §3.3, *Grafana OnCall*[17], an open-source tool, is used alongside the in-house *Grafana* instance to alert operators of any problems/errors with scripts experienced during a shift, or of any unexpected

[13] https://www.gnuradio.org/
[14] https://manpages.ubuntu.com/manpages/xenial/man8/rigctld.8.html
[15] https://manpages.ubuntu.com/manpages/xenial/man8/rotctld.8.html
[16] https://github.com/daniestevez/gr-satellites
[17] https://grafana.com/products/cloud/oncall/

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

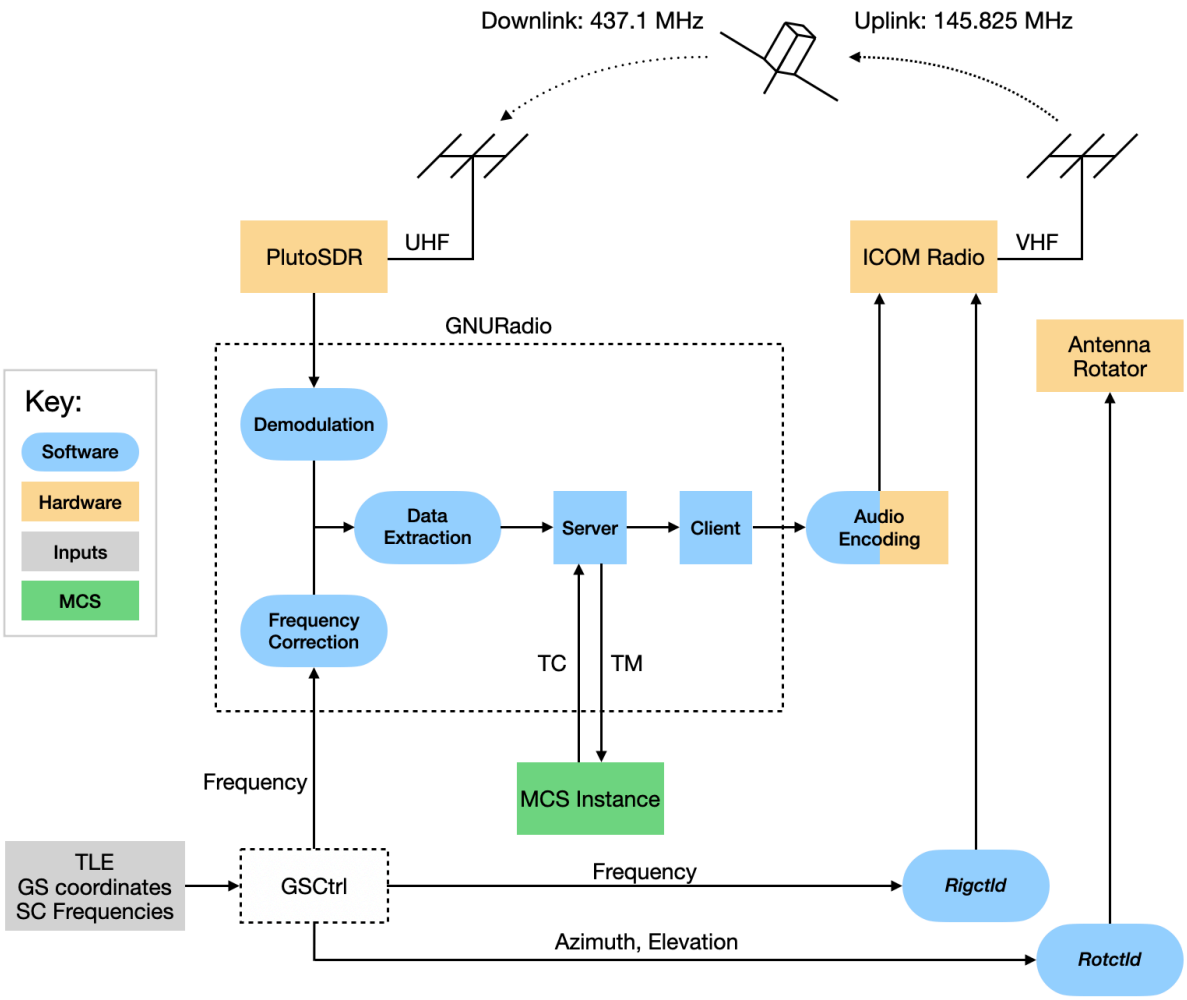

Figure 4: Overview of the interaction between hardware and software for automated ground station passes. Audio encoding is completed by hardware in BFD and by software in DQN.

spacecraft telemetry received. This service works with the *Grafana IRM* mobile application and can alert operators who are on shift. The on-call schedule is taken from the operations rota calendar, and this allows the service to ping the correct operator. Notifications can be sent with different priorities, customisable to each operator as desired, and with the possibility to alert through 'do not disturb' mode on phones. This tool has proven invaluable to the team, especially in the context of automated operations. Previously, an operator had to be present for all contacts to be aware of the ongoing health of the spacecraft, or if a shift had to be missed due to availability of the team, any problems with the mission were not known by the team until the next shift with an operator present. However, automated operations can now occur without the need of constant supervision, and operators can be made aware of any problems experienced only if necessary.

If an anomaly is encountered during shift hours from either an automated contact from one of the EIRSAT-1 ground stations or a beacon received from another ground station, the escalation chain of the notification to the operators is as follows:

1. A default notification is sent to the operator on call. This does not alert through 'do not disturb'.
2. After five minutes, an important notification is sent to the operator on call. This does alert through 'do not disturb'.
3. Ten minutes after this, a default notification is sent to the full team.
4. Fifteen minutes after this, an important notification is sent to the full team.

Figure 5: Overview of the interaction between hardware and software for automated ground station passes. Audio encoding is completed by hardware in BFD and by software in DQN.

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

If at any point a notification is responded to, the escalation stops, and no further notifications are sent for that anomaly. No notifications are sent between 12 am - 8 am UTC to allow operators to rest. Any notifications generated during this time are delayed until 8 am UTC.

Alerts are created when certain predefined conditions are met. These are stored as rules in the *Grafana* interface and can be easily created or edited. For EIRSAT-1, rules are based upon the values of certain parameters taken from either downlinked data or beacons. For example, safe mode corresponds to the parameter 'mission.ModeManager.mode' = 4, so an alert is sent when this condition is met. Due to the nature of mode transitions onboard, safe mode cannot be exited autonomously and hence this parameter will not change unless done so by an operator. However, some other rules which create alerts, for example a low battery voltage, could return to nominal ranges without need of human intervention. For these cases, a threshold for which the alert should stop firing can be set. This ability also proves useful for parameters which increment and wrap, such as the onboard uptime value. Using this nominal range logic, if the change in consecutive uptime values is negative, it means a reset or a wrap has occurred, and when the change becomes positive, the alert can stop firing as this is a once off event and not a continual problem.

6. Results & Conclusions

The suite of tools which has been and continues to be developed by the EIRSAT-1 team has made operations much more manageable with a small student-led team. Using open-source tools locally hosted on an in-house server has allowed tools to be tailored for our mission and operations. Since implementing automated ground operations, operators have needed much less time before and after each contact to organise, analyse and report on the results. As well as this, automated operations can carry out procedures with less overhead time, hence leading to more data downlinked compared to manual passes. With automated operations, all older spacecraft data (Event, HK, PASCAL, ADCSHK, and TED) have been fully downlinked, repeatedly causing the script to sit idle after completing as there is no more data to be downlinked. In terms of data acquisition from the spacecraft, this is the most optimal case, with maximal spacecraft data saved on ground storage.

While EIRSAT-1 is a single spacecraft mission, these tools demonstrate the possibility of automated ground operations for much larger missions, such as cubesat swarms whose contacts could occur around the clock. Due to the number of possible contacts for these missions, it can be very difficult to operate manually. In this case, ground operations with an alert system provides an excellent solution to carry out nominal operations autonomously, while alert operators when human intervention is required. This automation could increase the chances of success for any mission.

Acknowledgements

The EIRSAT-1 project is carried out with the support of ESA's Education Office under the *Fly Your Satellite! 2* programme. The authors acknowledge the guidance from Jean-Philippe Halain of the ESA PRODEX Office. The authors acknowledge all students who have contributed to EIRSAT-1 and support from Parameter Space Ltd. J.F. acknowledges support from the European Space Agency (PRODEX) (Grant No. 4000138314). D.M. (David Murphy), J.T., C.M., L.C., A.E., P.D., R.D. and M.D. acknowledge support from the Irish Research Council (IRC) (Grant Nos. GOIPG/2014/453, GOIPG/2014/684, GOIPG/2024/3488, GOIPG/2022/1008, GOIPG/2023/4396, GOIPG/2023/3741, GOIPG/2019/2033 and GOIP/2018/2564), respectively. C.M., A.E., and J.R. acknowledge support from the UCD Physics Scholarship in Research and Teaching. G.F. acknowledges support from a scholarship associated with the UCD Ad Astra fellowship program. S.M. and D.M. (David Murphy) acknowledge support from Science Foundation Ireland (Grant No. 17/CDA/4723). C.B., S.M., F.G.K., and D.M. (David McKeown) acknowledge support from Science Foundation Ireland (Grant No. 21/FFP-A/9043). L.H. acknowledges support from SFI (Grant No. 19/FFP/6777) and support from the EU H2020 AHEAD2020 project (Grant No. 871158).

Copyright ©2025 by University College Dublin. Published by the Canadian Space Agency (CSA) on behalf of SpaceOps, with permission and released to the CSA to publish in all forms.

Appendix A: EIRSAT-1 Channel Content downlinked in Automated Operations

Channel	Logging Period	Contents*
Event Log	N/A	Any events raised on board, such as (1) mode changes, (2) GPS locks and losses, (3) battery related events (i.e. beacon rate increasing and decreasing with high and low battery level, respectively), (4) TM and TC related events during passes, etc.
Housekeeping (HK)	50 seconds	Vital information, including (1) software modes and images, uptime, event count, (2) component temperatures, (3) battery voltage and current draw, (4) solar array temperatures, voltages and currents, (5) spin rates, (6) payload modes and error counters.
Telemetry Enhanced Diagnostics (TED)	10 minutes	Error-related data, including (1) Internal OBC errors, (2) Connections between OBC and payloads errors, (3) External communication errors, (4) Internal payload errors, (5) Battery error statuses, (6) Other miscellaneous error data, such as mode change errors or dropped events.
Power And Spacecraft Current And voltage Levels (PASCAL)	60 seconds	Power-related data, including (1) Solar panel power generation, (2) Spacecraft switch states and power consumption, (3) Spacecraft-wide voltages and currents, (4) OBC power usage, (5) ADCS power consumption.
Attitude Determination and Control System HK (ADCSHK)	5 minutes	ADCS-related data, including (1) ADCS power consumption, (2) Magnetorquer and gyro temperatures, (3) ADCS error counts, (4) Vector updates data.

* These are examples, and not exhaustive lists.

References

Doyle, Maeve, Rachel Dunwoody, Gabriel Finneran, David Murphy, Jack Reilly, Joseph Thompson, Sai Krishna Reddy Akarapu, et al. 2022. “Mission Test Campaign for the EIRSAT-1 Engineering Qualification Model.” *Aerospace* 9 (2). https://www.mdpi.com/2226-4310/9/2/100.

Dunwoody, Rachel, Maeve Doyle, David Murphy, Gabriel Finneran, Derek O'Callaghan, Jack Reilly, Joseph Thompson, et al. 2023. “Development, description, and validation of the operations manual for EIRSAT-1, a 2U CubeSat with a gamma-ray burst detector.” *Journal of Astronomical Telescopes, Instruments, and Systems* 9 (3).

Murphy, David, Alexey Ulyanov, Sheila McBreen, Maeve Doyle, Rachel Dunwoody, Joseph Mangan, Joseph Thompson, Brian Shortt, Antonio Martin-Carrillo, and Lorraine Hanlon. 2021. “A compact instrument for gamma-ray burst detection on a CubeSat platform I.” *Experimental Astronomy* 52: 59-84.

Murphy, David, Alexey Ulyanov, Sheila McBreen, Joseph Mangan, Rachel Dunwoody, Maeve Doyle, Conor O'Toole, et al. 2022. “A compact instrument for gamma-ray burst detection on a CubeSat platform II.” *Experimental Astronomy* 53: 961-990.

Murphy, David, Joe Flanagan, Joseph Thompson, Maeve Doyle, Jessica Erkal, Andrew Gloster, Conor O'Toole, Lana Salmon, Daire Sherwin, and Sarah Walsh. 2018. “EIRSAT-1 - The Educational Irish Research Satellite.”

Sawyer, D. M., and J. I. Vette. 1977. *AP-8 trapped proton environment for solar maximum and solar minimum.* Technical Report, National Aeronautics and Space Administration, Greenbelt, MD (USA). Goddard Space Flight Center.

Vette, J. I. 1991. *The AE-8 trapped electron model environment.* National Aeronautics and Space Administration, Greenbelt, MD (USA). Goddard Space Flight Center.