Space Sci Rev (2020) 216:8
https://doi.org/10.1007/s11214-019-0632-8

A Robust Framework for Low-Cost Cubesat Scientific Missions

In-Orbit Recovery, Results and Lessons Learned from UNSW-EC0

Joon Wayn Cheong[1] · Benjamin J. Southwell[1] · William Andrew[1] · Elias Aboutanios[1,2] · Chung Lam[1] · Tom Croston[1] · Luyang Li[1] · Shannon Green[1] · Alexander Kroh[1] · Eamonn P. Glennon[1] · James Bultitude[1,3] · Tim Broadbent[1,4] · Timothy B.Q. Guo[1] · Joerick G. Aligno[1] · Andrew G. Dempster[1] · Barnaby Osborne[1,5]

Received: 16 December 2018 / Accepted: 26 December 2019
© Springer Nature B.V. 2020

Abstract Cubesats have been effective at lowering the barriers for entry to space for educational institutions and small private players resulting in new and innovative missions and concepts. Novel, potentially powerful, space science projects such as QB50 can now be undertaken with limited budgets and resources. However, the failure rate of Cubesats has been quite high with many failing to establish any communications at all, leaving little opportunity for teams to debug and recover the satellite. Due to the time and cost restrictions faced by Cubesat projects, traditional verification and validation testing processes are not feasible, giving rise to the high failure rate. In this paper, we describe the experience gained during the development, launch and operation of the UNSW-EC0 Cubesat, which was deployed in 2017 as part of the QB50 mission. In particular, we present a robust framework derived from Failure Mitigation Effects and Criticality Analysis (FMECA) for Cubesat testing that is practical for typically resource and time constrained missions. We also describe robustness testing performed during development combined with additional functionality that was built into the satellite, which allowed in-orbit troubleshooting and mission recovery. Following its recovery, UNSW-EC0 was able to perform nominally for the remaining duration of its lifetime. Some preliminary in orbit mission results are also described in the paper. Two UNSW-built Single Event Upset (SEU) resistant experiments as well as the RAMSES payload successfully demonstrated long endurance operations in orbit.

Multi-Point Measurements of the Thermosphere with the QB50 Mission
Edited by David Miles, Robert Wicks and James Burch

✉ J.W. Cheong
cjwayn@unsw.edu.au

1 Australian Centre for Space Engineering Research (ACSER), University of New South Wales (UNSW), Sydney, NSW 2052, Australia

2 School of Electrical Engineering and Telecommunications, University of New South Wales (UNSW), Sydney, NSW 2052, Australia

3 Orbit Fab Inc., San Francisco, CA 94103, USA

4 Optus Satellite, Sydney, NSW 2085, Australia

5 International Space University, Strasbourg, Illkirch 67400, France

Published online: 20 January 2020

Keywords Cubesat · QB50 · Robust · Testing · FMECA · Recovery · Redundancy · UNSW-EC0

1 Introduction

The Cubesat standard has proven effective at enabling low budget space missions and putting space within reach of many educational institutions worldwide (Woellert et al. 2011; Selva and Krejci 2012; Toorian et al. 2008). Commercial-Off-The-Shelf (COTS) components designed to the Cubesat standard allows medium-ware providers to manufacture subsystem hardware at a larger scale, hence lowering the production cost and promoting miniaturisation of space hardware whilst improving the robustness of the individual components themselves (National Academies of Sciences, Engineering, and Medicine 2016).

The low-cost and short development cycle of Cubesats has led to their increasing application to scientific and Earth observation missions (Waydo et al. 2002; Blum et al. 2013). Although Cubesats are limited in size and power compared to large satellites, they are rapidly finding their niche in science missions that require multi-point observations and involve high mission risk (Poghosyan and Golkar 2017). This was indeed the rationale behind the QB50 project that called for a large constellation of small satellites for multi-point in-situ measurements of the lower thermosphere (Muylaert et al. 2009). The low altitudes dictate short orbital lifetimes that generally preclude the use of large, expensive satellites with the exception of those that carry on-board propulsion to maintain altitude, e.g. the GOCE satellite (Drinkwater et al. 2006). However, in this case, the availability of on-board propulsion means that the orbital lifetime is no longer constrained by the low altitude. Other exceptions may include scientific or other specialized missions where the mission objectives justify the large expense and short duration.

Despite the rapid rise in their popularity for serious scientific missions, Cubesats are notorious for their high failure rate. The high Cubesat failure statistics have led many to doubt their usefulness for carrying out space science. Small satellites, below 10 kg, are subject to a higher infant mortality rate (Guo et al. 2014) and many Cubesats fail to establish any communications at all. This leaves little opportunity for teams to debug and recover the satellite if it were possible. Unsurprisingly, the On-Board Computer (OBC), Electrical Power (EPS) and Communications (COM) subsystems cause the majority of failures in Cubesats (Langer and Bouwmeester 2016). Up to 66% of all Cubesat mission failures occur at a system level and 27% of them are due to a configuration or internal communications interface issue between hardware (Swartwout 2013). Due to the time and cost restrictions usually imposed on many Cubesat missions, traditional verification and validation testing (European Cooperation for Space Standardization 2010) and analyses (European Cooperation for Space Standardization 2009) are not feasible, thus directly contributing to the high failure rate.

In addition to continual advances in Cubesat hardware, improved mission development and execution processes are needed for Cubesats to realise their potential to deliver space science results. Therefore, it is the aim of this paper to contribute to the improvement of Cubesat mission robustness using the experience gained through the development, testing, launch and in-orbit operation of the UNSW-EC0 satellite that was part of the QB50 constellation. It is hoped that the testing framework and reasoning provided in this paper will especially benefit Cubesat builders attempting their first space science mission.

The paper makes a number of contributions.

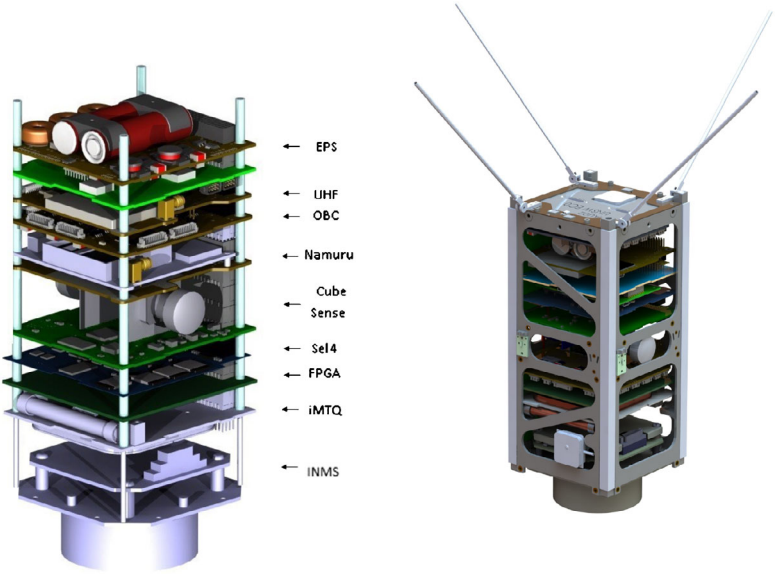
(a) Firstly, we address the robustness issues faced by Cubesat missions and suggest solutions to many, often unforeseen, complications that occurred during the UNSW-EC0 mission development.

(b) Secondly, we describe the in-orbit failure analysis and recovery technique that allowed us to rescue UNSW-EC0 after it initially failed to establish contact.
(c) Thirdly, we present an improved testing framework that can scale according to the resources available to the Cubesat teams. To this end, Table 2 that summarises key tests with risk-relevant metrics such that it becomes obvious to the Cubesat project managers what the consequence of scaling down various pre-launch tests will be. This serves as a quick-reference guide for Cubesat missions that are resource/time constrained and/or adopting the "fail early" mantra of agile methodologies.
(d) Finally, we propose new robustness tests (RBT) that we summarise together with the conventional Cubesat systems-test presented in Table 2.

2 UNSW-EC0 Mission

Educational Cubesat 0 (EC0)—also known as UNSW-EC0—is a Cal Poly standard (Heidt et al. 2000) compliant 2 unit (2U) Cubesat (see Fig. 1 and Fig. 2) that was developed and integrated by a team at the University of New South Wales (UNSW), Sydney. It is based on a variety of COTS components in addition to in-house designed and manufactured hardware. The project was established as UNSW's contribution to the QB50 project to conduct the most comprehensive study of the lower thermosphere to date (Osborne et al. 2013). Thus, UNSW-EC0 was designed to carry one of the QB50 science payloads, namely the Ion Neutral Mass Spectrometer (INMS). In addition to the INMS, the UNSW team used the remaining space to launch a number of experimental engineering and science payloads. The UNSW-EC0 payloads and their associated mission goals are:

– The INMS: This is an electrostatic particle analyser that was provided by the Mullard Space Science Laboratory of University College of London (MSSLUCL). It measures the neutral and single ionised states of O_2, NO and N_2 in the thermosphere. Along with a number of other QB50 satellites, UNSW-EC0 was intended to take unique and extensive measurements of the ionospheric composition to improve understanding of the variability of atmospheric drag, the chemistry of the thermosphere and impact of space weather on the upper-atmosphere.
– RUSH: The Rapid recovery from Single Event Upsets (SEUs) in reconfigurable hardware board aimed to test a number of algorithms for enhancing the robustness of satellite computing resources to radiation. Having effective recovery approaches that minimise device down time allows cheaper non-radiation hardened hardware to be used while providing the reliability needed for science missions.
– Kea Space GPS receiver: This UNSW developed secondary payload was to conduct two GPS-based remote sensing experiments: reflectometry and occultation. In the first instance, Kea was to investigate the use of GPS signals reflected off the sea surface to infer the sea state. The second goal of this part of the mission was to use Kea to analyse the refracted GPS signals passing through the ionosphere which would serve in space weather prediction.
– seL4bit: This is a formally verified micro-kernel that is guaranteed to have a worst-case execution time and is therefore suitable for real-time and mission-critical tasks, such as attitude determination and control. Flying seL4 on-board UNSW-EC0 was intended to give it flight heritage and test its operation and fault tolerance capability in the harsh environment of space.
– RAMSES: Rapid Manufacture of Space Exposed Structure (RAMSES) was an experiment that aimed to test the use of a world's first electroplated thermoplastic for the rapid and low-cost development of satellite structure itself using 3D printing. The satellite struc-



(a) interior (b) exterior without solar panels

Fig. 1 An interior/exterior CAD render of the UNSW-EC0 2U Cubesat. (**b**) shows deployable UHF antennas, antenna deployment PCBs and base plate with integrated Electrical Ground Support Equipment (EGSE) ports

ture was 3D-printed using Selective Laser Sintering (SLS) of nylon and then electroplated with nickel for thermal and electrical conductivity. This experiment was successful as the satellite was healthy and operational for the entire period until its end of life.

The list of payloads described above shows that UNSW-EC0 had a serious multi-faceted science mission despite its small budget, resources and size. The satellite was launched on 18th April 2017 and deployed from the International Space Station (ISS) on 2nd May 2017. It, however, failed to establish contact and its beacon was not heard (Southwell and Cheong 2017; Aboutanios and Cheong 2017; Cheong et al. 2018). After carefully analysing the situation, it was hypothesised that UNSW-EC0 had failed to deploy its antennas. The communication link losses that are due to the stowed antennas prevented the link from being closed. This allowed the team to devise a recovery plan involving a set of commands that were verified on the Engineering Model that was set up in the lab. To deliver these commands to the satellite, a GS with a significantly larger Effective Isotropic Radiated Power (EIRP) was needed to close the communications link. In what follows, we review the satellite architecture and then describe the failure mode analysis and associated process that permitted the recovery of the satellite.

3 UNSW-EC0 Architecture

The mission description, design, implementation and testing of the UNSW-EC0 Cubesat has been largely described beforehand (Cheong et al. 2016). UNSW-EC0 utilised COTS components from multiple vendors, primarily GOMspace, for much of the satellite stack. In total, the UNSW-EC0 bus was comprised of (see Fig. 3)

– NanoMind A712D On-Board Computer (OBC)—GOMspace

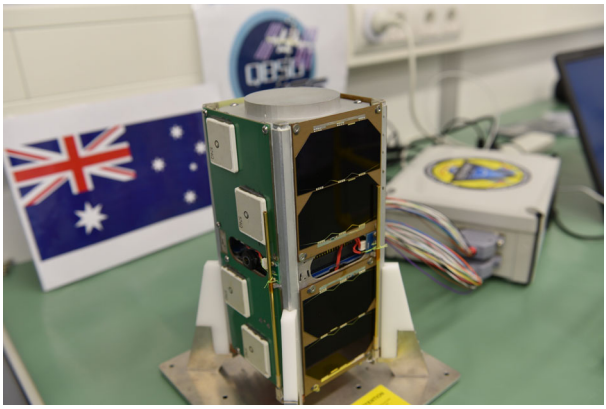


Fig. 2 A photo of UNSW-EC0 2U Cubesat in a stowed configuration. Photo taken during pre-integration checkout. The EGSE equipment in on the right

- NanoPower P31 Electrical Power System (EPS)—GOMspace
- NanoPower P110 solar panels—GOMspace
- NanoCom U482C Ultra High Frequency (UHF) Transceiver (COM)—GOMspace
- NanoCom ANT430 Omnidirectional UHF Antenna—GOMspace
- NanoHub IO Expansion (HUB)—GOMspace
- UNSW-EC0 AUXiliary Board (EAUX) (Cheong et al. 2016)—UNSW, Sydney
- RAMSES 3D printed Structure—UNSW, Sydney
- CubeSense Earth and Sun Cameras—ESL
- iMTQ 3 axis magnetorquer—ISIS

In addition to this, there were also the following experimental payloads

- Ion and Neutral Mass Spectrometer INMS—Von Karman Institute (VKI)/University College London (UCL)
- KEA GPS receiver (Glennon et al. 2011)—UNSW, Sydney
- RUSH (Cetin et al. 2016)—UNSW, Sydney
- seL4bit Microkernel Experiment (Data61 2018)—UNSW, Sydney

All components were integrated on a single Inter-Integrated Circuit (I^2C) serial bus (NXP Semiconductors 2014). As the bus was comprised of mostly GOMspace COTS products, the Cubesat Space Protocol (CSP) (GomSpace 2011) stack was utilised. This protocol supports routing operations, i.e., layer 3 of the Open Systems Interconnect (OSI) model (Zimmermann 1980), so that any module on the satellite bus can be directly addressed from the ground.

The EAUX board hosted a secondary OBC, a NAND flash memory for a backup file system, additional sensors for the Attitude Determination and Control System (ADCS), additional serial links and General Purpose Input Outputs (GPIO). Furthermore, the CSP stack was implemented on EAUX meaning that should the primary OBC fail the satellite could continue downlink experimental data collected by the onboard backup file system.

UNSW-EC0 hosted the Kea GPS receiver as a secondary payload. It is the latest in a family of FPGA-based GPS receivers designed by the Australian Centre for Space Engineering Research (ACSER) at UNSW. Kea was preceded by the Namuru series of receivers, the third generation of which was designed for Cubesat navigation (Parkinson et al. 2011; Choudhury et al. 2012) and specialized applications such as timing (Glennon et al. 2013). At time of writing this paper, Namuru V3 is still in orbit operating as avionics for the SHARC mission

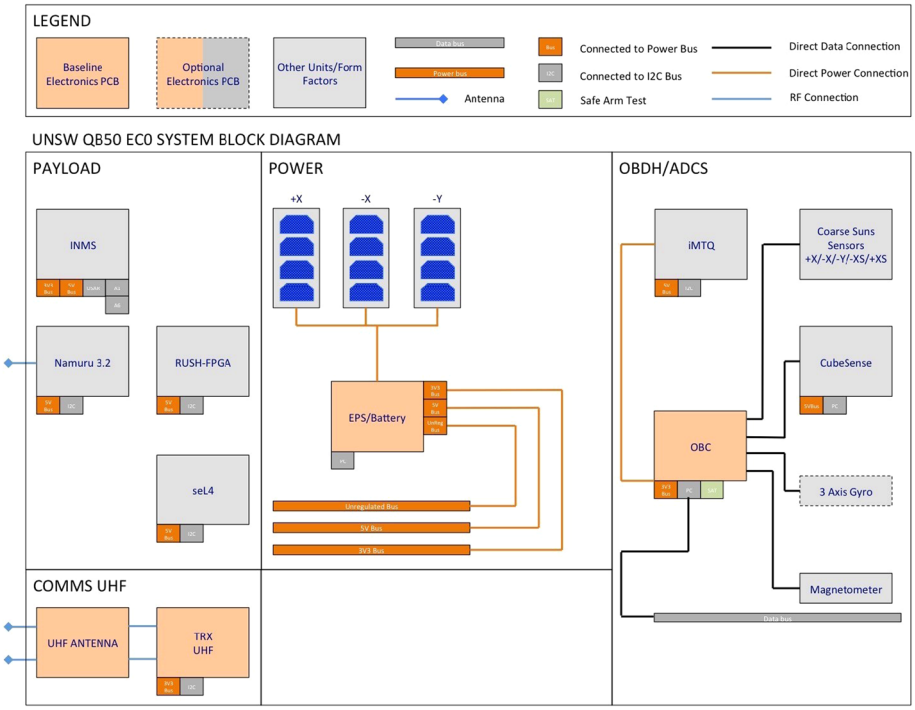


Fig. 3 Systems diagram to illustrate dependency and connectivity between various subsystems

(Glennon et al. 2013). Kea is a smaller but more powerful receiver that can be used for remote sensing tasks such as reflectometry and radio occultation in part due to its increased sensitivity (Glennon and Dempster 2016).

The use of the OSI model enabled GOMspace to develop File Transfer Protocol (FTP) client and server applications on top of the reliable user datagram protocol. Similarly, an additional protocol was built by ACSER on top of user datagram protocol for connectionless downloads of files from EAUX. This protocol was successfully utilised to download the relatively large photos taken by CubeSense. The simplicity of using the CSP stack also allowed us to pipe CSP traffic over the internet to a remote shell by replacing the RS-232 module with a pseudo-teletype port using the GNU/Linux `socat` utility.

A library provided by GOMspace allowed us to implement GOMspace Shell (GOSH) commands on the OBC. This allowed us to debug and control the satellite via a dedicated serial port. Instead of wrapping each command so that they could be executed with a specific CSP packet, a remote shell was implemented over CSP. This not only reduced the work required but also the risk of mistakes as nothing had to be implemented twice. This remote shell also lent itself to the development of a scripting language that enabled us to schedule GOSH commands to be executed at any time not just when a remote shell (i.e. active communications link) was available.

3.1 Power Systems Functionality

As depicted in Fig. 4, the EPS was able to transition between two states, on and off, depending on the battery voltage V and the predefined threshold V_{crit}. Thus, the EPS was

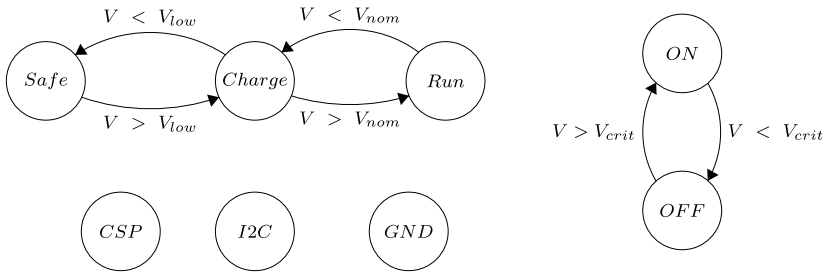


Fig. 4 State machine of OBC (left) and EPS (right). Note the I^2C, CSP and ground contact (GND) watchdogs power cycle the satellite if these bus transactions are not made within a designated time frame. These transitions are not shown for brevity. Note that the inequalities in the state machine transitions have a hysteresis so that rapid cycling between states does not occur

configured in such a way that it would only handle the case where battery voltage dropped below a critical level and the satellite needed to be switched off. The EPS default power-on behaviour can also be user-configured.

3.2 Antenna Deployment Functionality

There were four stowed antenna elements comprising the turnstile UHF antenna that needed to be deployed. The antenna deployment mechanism consisted of four GOMSpace-prescribed thin flexible Dyneema burn wires that were threaded through a Printed Circuit Board (PCB) with burn resistors and over each antenna element. Each wire was ensured to sit on two redundant burn resistors connected to two independent Single Pole Double Throw (SPDT) switching circuits. Contact between the burn wire and the burn resistor was maintained via the tension from the spring-loaded Nanocom ANT430 antennas. When triggered, the OBC's Initial Antenna Re-deployment (IAR) or Redundant Antenna Redeployment (RAR) sequence would command Nanohub's SPDT switch to short the burn resistors directly to the batteries, heating up the burn resistors which would then rapidly melt the burn wire.

The OBC was programmed to handle the deployment logic and also, for simplicity, the transitioning between safe mode, power charging mode and nominal run mode. Thus, the OBC would be switched on by the EPS and the OBC logic would then dictate if payloads or subsystems were to be switched on based on the battery voltage. From Fig. 4, we observe that the On and Off state are triggered by the EPS, whereas all other states are internal to the OBC. Thus, the battery voltage V dictates the predefined set of OBC tasks to be enabled or disabled. This software implementation was called the "task supervisor".

To conform with launch provider's requirements, specifically the delay after deployment before powering on UNSW-EC0 for the first time, both the EPS and OBC were required to control the satellite's operational mode. The "first boot" logic in the OBC:

- Disables all other tasks for 30 minutes counting down from the moment the OBC is powered on.
- Performs the IAR sequence at the end of the 30 minute countdown.
- Sets the non-volatile DPLYD flag at the end of the 30 minute countdown.

The above is also known as the "Boot" state in Fig. 6. Note that the "Deployed" (DPLYD) flag is cleared prior to the satellite pre-integration checks. The "first boot" sequence transitions to one of three other nominal states, but will be skipped if the DPLYD flag has been set

(presumably automatically at the end of the "first boot" sequence execution in the past). This design ensures that the "first boot" sequence only executes once in the Cubesat's lifetime in space.

If the OBC is in the "Charge" or "Run" states only, the RAR sequence would execute repeatedly at a pre-defined repetition interval. The RAR sequence would be disabled only upon manually setting a non-volatile "RAR disable" flag. The "RAR disable" is also cleared prior to the satellite pre-integration checks. It was intended that the "RAR disable" flag be set, manually by an operator, only after reliable communications had been achieved in the commissioning phase of the mission operations.

Note that the battery voltage needs to be sufficiently high in order to produce adequate heat in the burn resistors for reliable antenna deployment. Thus, both IAR and RAR skip the antenna deployment sequence if the battery voltage V is not sufficiently high.

3.3 Communications Functionality

The NanoCom U482C transceiver operated at a centre frequency of 436.525 MHz in the amateur UHF band. In the GS, a NanoCom TNC1 (Terminal Node Controller) interfaces the ground control software, csp-term, to the satellite using CSP by modulating an audio signal with Gaussian Minimum Shift Keying (GMSK) before a Kenwood TS2000 transceiver provided the audio to radio frequency modulation. This software afforded us the ability for both manual and autonomous commanding of the satellite. Additionally, the satellite would autonomously broadcast a beacon every 30 seconds using CSP once antenna deployment was successfully recorded on-board.

4 In Orbit Failure and Recovery

UNSW-EC0, together with other QB50 Cubesats were flown to the International Space Station (ISS) on the 18th of April 2017 as part of the OA-7 ISS resupply mission via the Cygnus capsule on an Atlas V rocket. Following the successful docking of Cygnus with ISS, the QB50 Cubesats CHALLENGER, NJUST-1 and UNSW-ECO were deployed into Low Earth Orbit (LEO) on 24th May 2017 at 0525 GMT. Another Australian Cubesat I-INSPIRE II, together with KPI-SAU-1 and SNUSAT-1 were also deployed into LEO on 25th May 2017 at 0400 GMT.

After the deployment of UNSW-EC0, the Ground Station (GS) at UNSW Sydney attempted to make radio contact with UNSW-EC0 during its first pass by listening for its 30 second radio beacon and, lacking a valid observation, then sought a response to 'ping' commands. The same radio silence observation was made by all other radio amateurs around the world including GS teams of other QB50 Cubesats. There was also complete radio silence from UNSW-EC0's sister satellite I-INSPIRE-II which shared many core bus hardware and software components.

4.1 Analysis

An initial fishbone analysis, sometimes also referred to as a fault tree analysis (Larson and Wertz 1992), identified multiple viable root causes for the radio silence. The most obvious is some form of physical hardware failure of the satellite bus. This mode of failure is fatal and recovery is not possible so we did not spend any more time pursuing this branch.

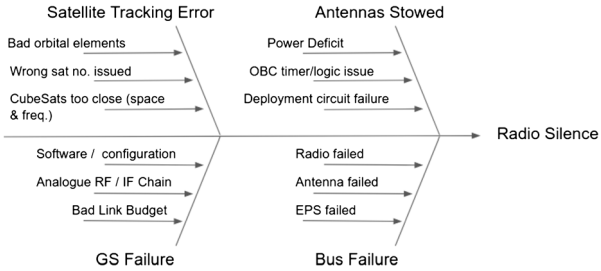


Fig. 5 Fishbone failure analysis of the failure to receive the UNSW-EC0 and I-INSPIRE II radio beacons

The next mode was actually a failure of the GS itself. However, UNSW GS could communicate with the engineering model using attenuators to provide a realistic signal level. This test resulted in the Low Noise Amplifier (LNA) being disconnected—which itself proved to have issues later on but these were not the reason for radio silence. Additionally, UNSW GS was successful in receiving beacons from GOMX-1 (Alminde et al. 2012) indicating that the antenna pointing, Doppler tracking and link budget were adequate. Furthermore, we were also able to track the beacons of different QB50 Cubesats (e.g. NJUST-1) and other GS were unable to receive the UNSW-EC0 beacon so a failure in the UNSW GS RF receive chain was ruled out.

It was still possible that UNSW GS could track (both antenna pointing and Doppler) GOMX-1 but not UNSW-EC0. A hypothesis was put forward that suggested the Cubesat's position computed using Two Line Elements (TLE)s supplied by Joint Functional Component Command for Space (2018) were incorrect. Specifically, either the orbital elements themselves or the satellite number issued to UNSW-EC0 being incorrect (i.e. identifying a different space object as UNSW-EC0) could also cause radio silence to be experienced by all GSs. Therefore, we tried the TLEs of CHALLENGER and NJUST-1 as they were released in the same deployment cycle as UNSW-EC0, but still failed to receive the expected beacon. As will be shown later, this was one of a number of contributing factors to the radio silence.

It was also hypothesised that the antennas did not deploy. While difficult to prove, it would cause radio silence due to the losses resulting from its stowed configuration. It was also the only possible root cause that can potentially be rectified so we pursued this branch.

Failure of the antenna deployment circuitry was plausible but unlikely. There were redundant burn circuits and burn resistors for antenna deployment. A complete deployment circuit failure, as part of the antennas stowed branch of Fig. 5 would lead to an irrecoverable failure that would be impossible to fix, hence this sub-branch was not pursued.

UNSW-EC0 would not begin to detumble until the deployment sequence was completed. Thus, if it had failed due to some OBC or EPS watchdog timer or logic issue then it was likely that there was also a power deficit. Additionally, the battery charge state dictates the operational mode of the OBC. For example, if it is too low then the OBC falls back into safe mode, i.e., the decoupled state machines presented earlier in Fig. 4 are oversimplified. To eliminate any interaction between the state machines of multiple subsystems that could be persistently preventing the deployment to execute, we coupled the EPS and the OBC battery voltage-dependant state machine diagrams into one, shown in Fig. 6.

One hypothesis in the OBC timer/logic issue is that, if the boot sequence is periodically interrupted or rebooted by, for example one of the watchdog resets, then it becomes plausible that the OBC will never be able to complete the boot sequence. Subsequently, the IAR sequence will never be executed as the next reboot will restart the 30 minute countdown. In this hypothesis, it is the indefinite rebooting into the Boot state that is the predominant factor in the observed failure. If IAR were bypassed and the OBC were to enter any one of three

nominal states, the RAR could still be depended on for antenna deployment. Solution 1 of Sect. 4.2 resolves this.

Elaborating further on the "Power Deficit" sub-branch of the "Antenna Stowed" failure, prior to the Cubesat's delivery, we did test the antenna deployment mechanism under cold temperatures and low State of Charge (SoC). At 0 °C, the success rate of one IAR deployment sequence was found to be less than 50%. At very low SoC ($\approx$ 7.2 V), we were unable to achieve reliable deployment at all. However, at room temperature and above nominal SoC ($\approx$ 7.8 V), we were able to achieve 9 out of 10 successful deployment. To enhance the chance of success, the RAR sequence was designed to execute indefinitely, thus ensuring that the deployment sequence would occur at various parts of the orbit (thus possibly coinciding the deployment with parts of the orbit where the Cubesat was exposed to a greater amount of Sun's radiated heat). Assuming that the EPS and solar panels are functional and charging the batteries, the recurring RAR also ensures deployment would occur as long as there was an overall positive power margin that charges the battery up. We did verify that the recurring RAR consumes approximately 8% of the battery's capacity per day.

Assuming that the battery was at half capacity at insertion into orbit, another hypothesis that we could formulate was that, if the EPS were severely under-powered, it may dwell in the "Safe" most of the time and never allow the RAR sequence to execute. For this, we could increase the hysteresis between the "Safe" and "Charge" states such that a slowly charging EPS would dwell sufficiently in the "Charge" state to perform the RAR sequence. Regardless of the likelihood of this occurring, we wanted to be able to accommodate the most severe case of power deficit. For this, we could maximise the possibility of RAR execution by changing the voltage threshold, V_{crit}, for transitioning out of the "Off" state to a higher value. This is factored into Solution 2 in Sect. 4.2.

We had now identified multiple scenarios that would cause the deployment logic to fall into an infinite loop, continually power cycling in an attempt to charge the batteries. After identifying these states, we are able to design a command set that would force the logic to follow a deterministic path to recovery.

4.2 Recovery

4.2.1 Designing the Composite Recovery Sequence

The state machine in Fig. 6 relies on non-volatile flags written to the SD card and battery thresholds written to the flash memory to control the satellite. The following command sets were produced to force the non-volatile flags into a state that would provide a SoC that would allow the deployment logic to execute:

1. Skipping the boot sequence would also bypass the IAR but it would not bypass the RAR routine. The solution would be to set the DPLYD flag so that it would not enter the Boot state and periodically execute RAR as designed.
2. Additional commands were constructed to set the hysteresis on the battery charge to be as large as possible (potentially up to 48 hours) so that there would be enough energy available for a reliable burn. The restrictions imposed by the mission envelope meant that these parameters were not already set to these values.
3. To cater for other unforeseeable failures, we also directly commanded the HUB to deploy the antennas. This command bypasses the OBC.

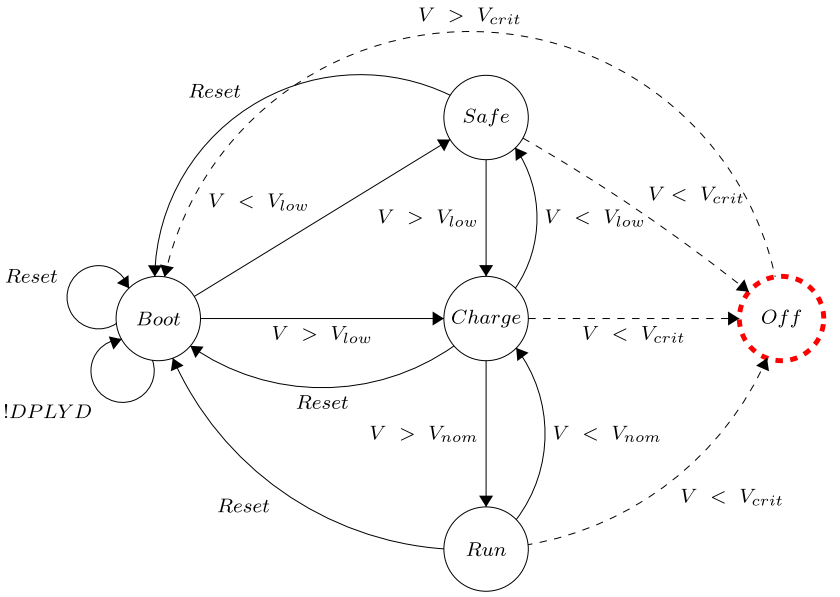


Fig. 6 State machine considering both OBC and EPS. The transitions occurring as a result of the EPS are shown as dashed lines. The reset transition is also shown and can be caused by a watchdog, system crash or a command. The $!DPLYD$ is a non-volatile flag written once the deployment sequence was executed. If it failed to execute the OBC would not leave boot mode. Again, the inequalities in the state machine transitions have a hysteresis so that rapid cycling between states would not occur. These are not shown for clarity

4.2.2 Recording of Ham-Compatible Uplink Sound Bites

The CSP communication model is very similar to that of the AX.25 amateur packet radio system. The GOMSpace Nanocom U482C communications subsystem uses GMSK modulation. The data stream is modulated using Audio Frequency Shift Keying (AFSK) audio frequencies and then upconverted to radio frequency using frequency modulation (FM). Radio amateurs frequently use software defined TNCs and so already have a hardware link between a PC and transceiver via a soundcard for packet radio operation. This meant we were able to record the command set using common audio recording equipment and send it to the operators who could then easily replay the recorded command set instead of using their own software defined TNC.

4.2.3 Overcoming Stowed Antenna Losses

The estimated additional losses due to the antennas being stowed were approximately at least 20 dB (based on email communications with GOMSpace engineers) which according to Table 1 would result in an insufficient link margin for uplinks from the UNSW GS. After several unsuccessful uplink attempts offered to us by several other amateur radio GSs (who did not have more than 20 dB of link advantage), we managed to engage the C.A. Muller Radio Astronomy Station (CAMRAS) team to uplink our commands. The relative gain of the CAMRAS station compared to the UNSW GS is 23.5 dB as shown in Table 1 and which would overcome the expected losses of the stowed antennas.

On 12 June 2017, the CAMRAS team who operate the 25 m Dwingeloo Radio Telescope in the Netherlands (Elbers 2017) transmitted the recovery commands to UNSW-EC0 and

Table 1 Uplink Link Margin improvement achieved to overcome the losses due to a stowed antenna

	UNSW GS	CAMRAS
Antenna Gain	15.5 dBi	30 dBi
Average Pointing Loss	−3 dB	−3 dB
Transmit Power	17 dBW	26 dBW
EIRP	29.5 dBW	53 dBW
Effective Link Margin Improvement		23.5 dB

I-INSPIRE II over its first pass. On its second pass, the beacon from I-INSPIRE II was successfully detected with nominal received signal strength. Because I-INSPIRE II shared the same hardware and software code that made up its deployment sequence, it indicated that the devised commands fixed the root cause of the problem. However, no beacons were received from UNSW-EC0 by the UNSW GS or other GS teams.

4.2.4 TLE Mix Up

The Two Line Elements (TLEs) which contain the orbit parameters of an orbiting object are issued by Joint Functional Component Command for Space (2018). The first TLE for UNSW-EC0 was issued on 26th May 2017, attributing UNSW-EC0 to the NORAD assigned identifier (NORAD ID) 42721. The TLE for NORAD ID 42721 had been used on 12th June 2017 for the uplink transmission of recovery commands.

Following the success of I-INSPIRE II's recovery and the failed recovery of UNSW-EC0 on 12th June 2017, our focus shifted to the TLE as the likely cause of UNSW-EC0's failed recovery. This speculation was further substantiated when it was found that CHALLENGER's GS team was using the TLE of NORAD ID 42721 for antenna pointing and Doppler correction to operate its Cubesat. The effect of a TLE mix up at this stage, where satellites of the same deployment cycle were not yet widely spaced out by differential drag, is sometimes marginal. This is especially true for low baud rate communications where low gain antenna is used. But the 25 m radio telescope dish with an unusually high gain has an extremely narrow beamwidth such that small antenna pointing errors would cause large signal attenuation. Therefore, it became necessary to retry the recovery procedures again using CHALLENGER's TLE (assigned NORAD ID 42723).

On 18th June 2017, the CAMRAS team attempted to uplink the recovery commands to UNSW-EC0 using this TLE. At its next pass, the CAMRAS team verified that UNSW-EC0's beacon was detected over a low elevation pass. On the same day, UNSW GS successfully received and decoded the UNSW-EC0 beacon with expected telemetry data. The telemetry indicated nominal health of the satellite, as shown in Fig. 7. The OBC timestamp (in UNIX format) was incorrect as the on-board Real Time Clock (RTC) had not yet been synchronised. It had lost its last time synchronisation from the ground as the RTC had been powered off beyond its designed timekeeping duration.

During the lifetime of UNSW-EC0, SatNOGS (2018) received 7280 decoded beacon packets, verifying that the Cubesat had been consistently broadcasting its beacon and operating nominally. Similarly, 4876 decoded I-INSPIRE II beacons were recorded by SatNOGS since its successful recovery. Reaching the end of its designed mission life, UNSW-EC0 re-entered Earth's atmosphere on 4th December 2018.

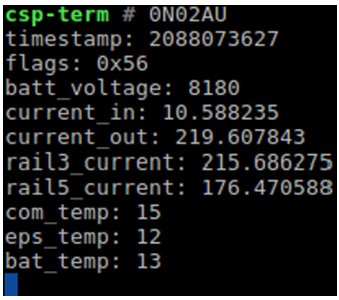


Fig. 7 The first beacon received from UNSW-EC0 by the UNSW GS. All data was nominal. The real time clock had not yet been synced so the timestamp is offset and the flags variable indicates that the correct power rails are switched on and the satellite has not yet been commissioned. This particular beacon is produced by the OBC so we could also infer that the primary I^2C bus, EPS, primary file system, solar panels and obviously COM were all operational

4.3 Root Cause of In-Orbit Autonomous Antenna Deployment Failure

It was intended that the OBC would be powered upon the insertion of UNSW-EC0 into orbit from the ISS. This is effected by three redundant deployment switches associated with the EPS subsystem. Upon OBC power on, a 30 minute delay was implemented before the deployment sequence, followed by nominal operations. As per Nanoracks' (the launch integrator) requirement, the OBC inhibits radio communications (including its periodic beacon) and turns off all other OBC functionalities during the 30 minute deployment countdown period. Following the successful recovery of UNSW-EC0, we downloaded past OBC logs to identify the root cause of failure. The first half of the log downloaded after recovery indicated that as the deployment counter was ticking down, the OBC rebooted every 10 minutes, resetting the deployment counter each time. This can be seen in Fig. 8. The unexpected OBC reboots were caused by the EPS watchdog monitoring the I^2C bus. This watchdog had been enabled to resolve occasional I^2C bus anomalies (see Sect. 5) that would otherwise render the I^2C inaccessible until a power cycle. The watchdog simply power cycles all subsystems if no I^2C transactions were detected within a 10 minute window. One of the essential OBC functions turned off during the 30 minute deployment window is the I^2C traffic generated between the OBC and the communications subsystem for the periodic radio beacon to be transmitted. Given the lack of any I^2C transactions, the I^2C watchdog would regularly power cycle all subsystems, including the OBC.

Extensive pre-shipment tests has been conducted to verify the programmed deployment sequence. However, the deployment sequence test procedure included periodic checks of the configuration and power state via the OBC diagnosis Universal Asynchronous Receiver-Transmitter (UART) port.

This periodic diagnostic procedure ensures that the test is on track according to our expectation. It was unforeseen that these checks triggered I^2C traffic that otherwise would not have occurred. Without hindsight, it was not expected that internal GOMSpace OBC processes would not generate any I^2C traffic at all during the 30 minute boot period. Processes such as task supervisor and CSP watchdog heartbeat signals (every few minutes) were verified to have triggered I^2C signals, but were not registered by the EPS as I^2C traffic, but rather as CSP traffic. Therefore, we concluded that the periodic OBC UART check had inadvertently triggered I^2C transactions that prevented the antenna deployment failure from presenting itself during pre-shipment tests.

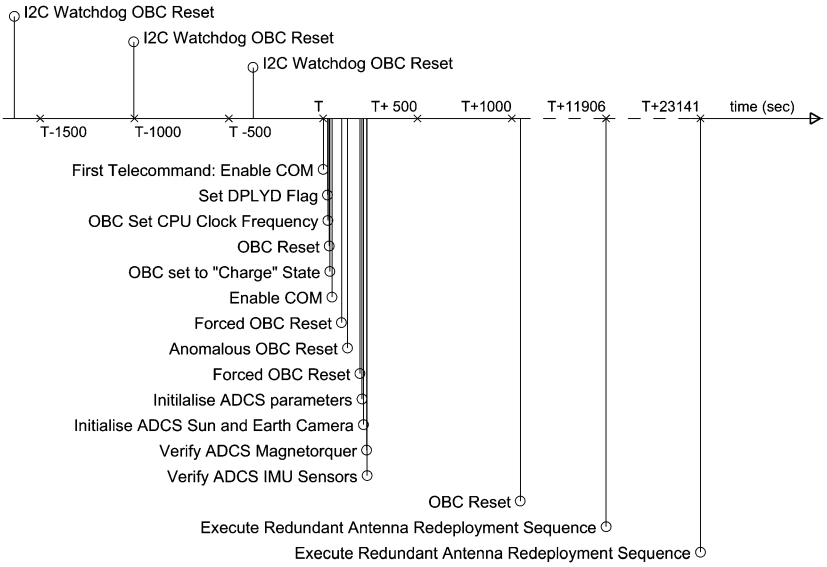


Fig. 8 The timeline of events before and after the time of receipt of the first telecommand at time, T. The timeline is based on the logs retrieved from UNSW-EC0. Note that there is intentional truncation of the timeline towards the right for better visualisation

5 Robustness Issues and Solutions with Cubesats

5.1 "First Boot" Issues

Specific to the topic of the 30-minute deployment delay, it can be argued that its implementation in the EPS would have prevented unnecessary complexities for the following reasons: (a) Nanoracks actually recommends that the 30-minute deployment delay be implemented at the hardware or EPS level to prevent logic-related erroneous antenna deployment inside the NanoRacks CubeSat Deployer. (b) The 30-minute deployment delay would not have been affected by I^2C watchdogs (which caused our Cubesat's failure) if implemented in the EPS. (c) The 30-minute deployment delay implemented in the EPS would have correctly prevented other modules (such as the COM subsystem) from turning on at a hardware level during the 30 minute window.

5.2 General Subsystem Issues

The OBC was delivered with source code for a barebones FreeRTOS project that was to be used as a starting point for firmware development. Additional libraries were also purchased that provided a file system which allowed us to develop logging capability on top of that. The file system was to be accessed by multiple threads asynchronously and, during development, concurrency issues were encountered that sometimes lead to corruption of the file system.

Additionally, there were also many functional requirements of the mission that the Cubesat had to meet such as the deployment delay and the telemetry required in the beacon. Ideally, these functions should have been handled by the EPS and COM respectively. However, they did not support that functionality and were closed source so the OBC was required to handle these functions, which greatly increased system complexity.

Cubesats stand to benefit enormously from free open source software such as GNU/Linux which provide much more functionality and flexibility in addition to the years of testing, maintenance and improvement that are contributed by the community. The KubOS framework (Plauché 2017) is an example of a GNU/Linux distribution designed for Cubesats. All of the core functionality is already implemented and a Cubesat team only needs to code mission specific logic. Many other potential open source software libraries from the Internet of Things (IoT) community may also be applied in the Cubesat context. Features from such community-driven code base are often robustly tested and can be easily manipulated to suit Cubesat applications with little processing overhead.

5.3 Radio Communications Issues

The use of amateur radio frequencies for Cubesat missions is popular due to the relatively low cost and the ease associated with becoming licensed for a Cubesat engineering team. This also permits many amateur radio operators to be able to receive beacons to support the mission. Indeed, the satellite frequency slots available for allocation are increasingly overcrowded and Cubesat managers are facing difficulties and delays in securing slots in recent years.

Additionally, the use of amateur radio bands means there are many COTS components available to a Cubesat team to develop a ground station with. At the time of procurement of the UNSW-EC0 components, the only architecture available was using a COTS TNC along with an amateur radio transceiver. However, most amateur radio hardware is designed for voice, not packet operation. Thus, the IF bandwidth of many transceivers limits the achievable system baud rate to below what the Cubesat radio and TNC are capable of. Additional tuning is also required, including adjusting the line levels of the audio link between the TNC and radio.

Furthermore, some components are simply not compatible with packet mode operation. For example, many LNAs are intended to be voice activated (VOX) and have tens of milliseconds of latency that makes half-duplex data communication difficult without introducing controlled link turnaround delays.

5.4 I^2C Bus Related Issues

The I^2C is a multi-master system that allows any device to become a master for a specific I^2C transaction. Clock stretching is a feature implemented in some I^2C devices that enables a slave to slow down the bus operations by holding the clock line low. During the development of UNSW-EC0, it was found that the GOMspace NanoCom radio (i.e. the COM subsystem) would hold the clock line low (i.e. clock stretch) during a particular I^2C operation between the OBC and the EAUX. In most cases this malicious and excessively prolonged clock stretching by the COM subsystem would not recover, thus requiring the I^2C watchdog on the EPS to power-cycle the bus after 10 minutes.

It was found that the COM subsystem would be triggered to clock stretch if it did not receive a specific non-mandatory I^2C end of packet symbol. To make matters worse, the clock stretching may occur even when the source and destination (or the master and slave of that particular I^2C transaction) of the packet do not involve the COM subsystem at all.

We were able to identify the anomalous COM clock stretching behaviour via a logic analyser. However, as the root cause lies within the firmware of the COM subsystem which we are unable to reprogram and is closed source, a workaround was necessary. Therefore,

we ensured that the OBC and EAUX I^2C packet will be appended with a non-mandatory end of packet symbol.

Many Integrated Circuits (ICs) have an I^2C interface and given its lack of external components (apart from pull-up resistors), its simplicity in hardware and its functional design makes a single I^2C bus attractive for fast development cycle missions such as Cubesats. However, as the I^2C bus is a single point of failure and exposed to experimental hardware on most Cubesats, it is not a robust design choice. In the case of UNSW-EC0, the limited COTS parts available during the procurement phase (circa 2015) rendered this to be the only viable architecture to our team. In comparison, the CAN bus is a relatively robust cousin of I^2C. The recent publication of the open source ECSS-CAN bus (Scholz et al. 2018) stack and the fact that some Cubesat COTS manufacturers are starting to offer the CAN bus as an option, are key factors in boosting the appeal of the CAN bus for future Cubesats.

Additionally, the main system's I^2C bus should be isolated from that of the payload's I^2C bus to reduce the severity and risk of a potential inter-subsystem communications and ground-to-space communications failure. This is especially true when payloads connected to the I^2C bus do not have flight heritage.

The I^2C specifications contain vagaries and ambiguities that allow various chip manufacturers to implement it such that, in certain conditions, subsystems from heterogeneous manufacturers may not fully cooperate with each other and occasionally result in a hung bus. In fact, we have experienced this issue and similar problems have also been reported on community forums. The CAN bus is more robust as it has been used heavily in automobile applications and has been advocated by some Cubesat developers and suppliers (Bouwmeester et al. 2017; Scholz et al. 2019).

5.5 ADCS Subsystem Issues

The ADCS was a bespoke system developed in house using COTS components. The ADCS had 5 modes of operation including a self test mode where data of all sensors and actuators were logged for debugging purposes on the ground. Logs of the magnetometer data were used to confirm detumbling had successfully completed before the satellite was commanded to enter pointing mode. When in pointing mode, if the Extended Kalman Filter (EKF) has not converged, the satellite continues to actuate as if it were in detumble mode.

It was soon discovered that the EKF was either never converging or was diverging quickly after the fact. After learning this, the satellite was commanded to log the output of the Earth and Sun vector estimation algorithm and it was found that the Earth vector was not being detected. The next step was to take a photo with the Earth camera to confirm it was in fact operational. This verified that the camera and associated hardware and algorithms were in fact operational but that a burn wire was floating in the field of view and which was disrupting Earth vector estimation algorithms that rely on edge detection. The COTS camera, CubeSense, supports masking regions in the image for this exact scenario but the mask is not saved into non-volatile memory. The OBC was programmed to store the CubeSense configuration in its own non-volatile memory so that the CubeSense configuration was persistent. However, it was not programmed in such a way that the image mask could be changed at run time.

The majority of the ADCS sensors reside on the main I^2C bus. The secondary magnetometer is the only sensor that does not as it resides on a separate point-to-point I^2C with the OBC. Additionally, the magnetorquers are driven by the OBC General Purpose Input Output (GPIO) directly. This architecture makes the detumbling mode immune to problems on the main I^2C bus, even if it completely fails. Logs collected over the duration of the mission indicate that the ADCS was required to switch to the secondary magnetometer multiple times.

Fig. 9 Photo taken by UNSW-EC0 on the 19th March 2018 at 378 km altitude. On the right hand side of the image, the burn wire that was stowing one of the antennas can be seen

The length of time the primary magnetometer was unavailable corresponds with I^2C bus crashes discussed in Sect. 5.4.

After delivery of the flight model, an engineering model was procured to enable on ground testing throughout the mission. However, due to the immaturity of the COTS Cubesat market nearly all components had hardware and firmware changes so exact replication of issues on the flight model was made extremely difficult.

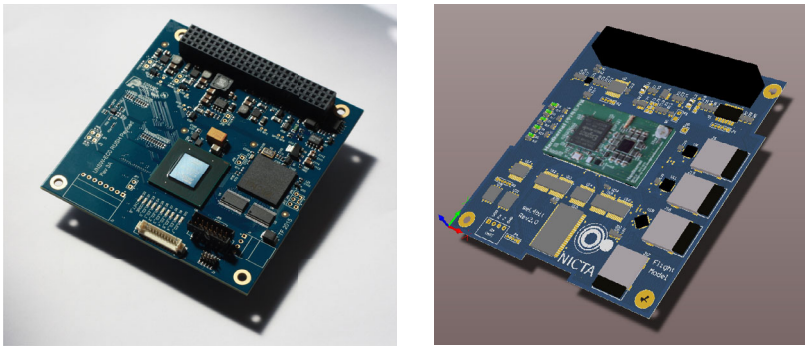
6 Mission Results

UNSW-EC0 and I-INSPIRE II are Australia's first ever pair of operational Cubesats in space. UNSW-EC0 was built solely by UNSW Sydney engineers while I-INSPIRE II was built by engineers and scientists from UNSW Sydney and the University of Sydney. After successful recovery, UNSW-EC0 went on to operate all of its payloads. Because the ADCS attitude estimation was unable to converge due to a lack of Earth and Sun vectors, the Kea GPS was not able to navigate as UNSW-EC0 was not pointing. The RUSH and seL4 experiments (see Fig. 10) were more successful: both of them passed self testing shortly after being commissioned and both were then able to perform their designated experiments. The seL4 secondary experimental payload had its experiment executed several times nominally as expected, one of which was able to continuously operate for more than two hours before it was turned off to conserve battery. The RUSH secondary experimental payload was able to be successfully operated for several hours cumulatively.

The EAUX board, which was developed at UNSW, gained flight heritage. The EAUX board was instructed directly from the ground to take the photo in Fig. 9. It did this via a point-to-point UART connection between EAUX and Cubesense and stored the photo on the backup file system located on EAUX. Then a simplified connection-less FTP stack made in-house was used to download the photo to the UNSW GS over a series of passes. The above results demonstrated the ability of EAUX to perform as an OBC on future missions.

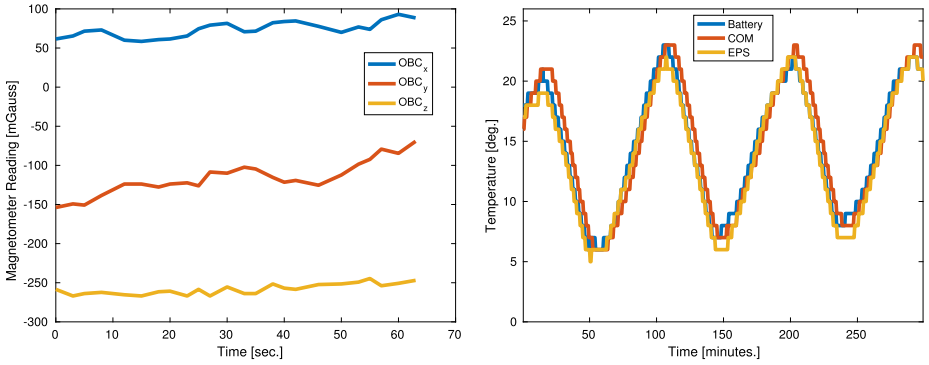
Even though the UNSW developed ADCS failed to perform pointing, detumbling was verified in sensor data logged not long after recovery, shown in Fig. 11(a). Further data, including multiple photos taken with CubeSense Fig. 9, collected over the duration of the mission will support the development and simulation of future ADCS systems (see Sect. 5.5 for more information).

The Kea GPS payload was consistently operational and responsive to all OBC commands throughout its mission life. However, due to the inability of the ADCS to accurately obtain



(a) RUSH (b) seL4

Fig. 10 (**a**) The RUSH experiment and, (**b**) the seL4 experiment. Both were developed by UNSW, Sydney and executed their respective experiments on UNSW-EC0 in 2018



(a) OBC Magnetometer (b) Bus Temperature

Fig. 11 (**a**): Magnetometer data logged to verify that detumbling had been completed. The ADCS utilised the B-dot algorithm so the angular rate will never converge to zero. (**b**): Temperature of the satellite bus over multiple orbits wherein diurnal effects can be seen. The temperature plots are indicative of good satellite health, in addition to nominal thermal operating points, the I^2C, OBC, EPS, COM, on-board storage and sensors demonstrate uninterrupted operations for days

attitude estimates itself, it was not possible to maintain the GPS boresight towards zenith over an extended period of time which severely affected Kea's ability to navigate. The GPS receiver operates in cold start mode which requires a persistent $\approx$ 20 minute window to acquire and track the minimum of four satellites to produce useful navigation results. While efforts were made to put the receiver into a warm start mode via the provision of external information such as millisecond level time synchronisation and the spacecraft's TLE, the receiver still had to obtain the GPS satellites' almanac on its own via continuous decoding of the GPS data for a duration exceeding the ten minutes necessary for the full almanac to be decoded. It is worth noting that the Kea GPS receiver has been proven to be capable of navigating in space in the Buccaneer risk mitigation mission.

The GS software was developed by UNSW Sydney (Southwell 2018; Brodie et al. 2017) and ran autonomously for most of the mission duration. This software, which now has mission heritage and was operational for over 99% of the time, will be used for future missions. Furthermore, over the duration of our mission, 27,000 packets had their RSSI levels recorded

by the radio onboard UNSW-EC0 with a mean level of −104 dBm validating the original link budget.

7 Robust Framework

To check for the compliance of complex space systems such as satellites, NASA and ESA have well documented and publicly accessible waterfall-style testing frameworks (Royce 1987). However, as mentioned previously, such testing frameworks are typically impractical for Cubesat missions that are resource and time constrained. Since testing and validation of Cubesats cannot be avoided, a scaled-down set of tests that would apply to a wide range of Cubesat missions depending on its allocated resource would prove very useful. This leads us to propose a scalable Cubesat testing framework in this section.

The Cubesat form factor is mostly chosen by low-budget space missions with strict time constraints for risk-mitigation purposes. Hence, failure mitigation is often overlooked or assigned a lower priority. However, there are key technical features from the UNSW-EC0 mission that demonstrated its robustness with minimal impact on resources. In Table 2, we show the minimal set of tests and, unlike many other testing frameworks, we also present relevant metrics to assist Cubesat project managers in assessing the risk and the likelihood of success of a Cubesat mission. The justification of the tests, value of metrics associated to the tests and the Cubesat configuration type are described in the subsections below.

Resource constrained Cubesat teams need to employ agile practices for maximum efficiency. One of the useful mantras of agile practices is to iterate and to fail early. This translates into performing system-level tests as early as possible. If it fails, then pursue the associated lower-subsystem or unit level test to isolate the root cause of failure to fix it. Once it is fixed, the system-level test is repeated. This principle can be applied in Unit Function Testing (UFT) and System Level Testing (SLT) during the development phase. However, once we enter the final testing phases of environmental tests (ENT), Simulated Deployment Tests (SDT), Day in Life Tests (DLT) and Robustness Tests (RBT) (see Table 2), this iterative method may not be practical as the failure of one test may likely invalidate the other.

The numbered tests in ENT and SDT of Table 2 should be executed sequentially. It is important to note that hardware and physical modification of any components after ENT invalidates all tests performed from ENT on. Hence, it is important to make re-programming of the OBC and other onboard micro-controllers possible without invalidating the environmental test.

Instead of being a formal validation process, the proposed robustness framework in Table 2 ensures that the majority of the Cubesat features are validated before flight. The framework also provides useful statistics associated with each sub-test (see Failure Mitigation Effects, and Criticality Analysis, FMECA methodology European Cooperation for Space Standardization 2009) so that the risk taken for bypassing a certain test is known.

The framework proposed in Table 2 is a combination of test processes practised in the UNSW-EC0 mission and test processes deemed necessary in hindsight. It is hoped that this framework would serve as a quick-reference guide to future Cubesat builders. Of the tests mentioned, UNSW-EC0 have performed various UFT and SLT, all of ENT, all of SDT, and DLT 01-04. These were all mandated by VKI and Nanoracks for launch qualification. We also performed RBT 01-05 to ensure robustness. On the other hand, DLT 05-06 and RBT 06-08 were not performed. From our experience, we believe passing these tests would substantially improve the likelihood of mission success, as indicated by the failure severity and failure likelihood of some of these tests. If these tests and features were able to be implemented and performed, many of the complications experienced in our mission could have been prevented.

Table 2 Minimal set of robustness tests applicable for flight readiness review, mission readiness review and verification Report. This table summarises the tests, their likelihood (Lik.), severity (Sev.) and purpose. The purpose of the tests include development (DEV), verification and validation (V&V), Physical Failure Prevention (PFP), Dead On Arrival Prevention (DOAP), Infant Death Prevention (IDP), maximise mission output (MAX) and Mission Failure Prevention (MFP)

Test ID	Test description	Type	Lik.	Sev.	Purpose
	Unit Function Testing (UFT)	PSPI			DEV V&V
	System Level Testing (SLT)	FSPI			DEV V&V
	Environmental Test (ENT)				
ENT 01	Vibration Test	FSPO			PFP
ENT 02	Thermal Vacuum Test	FSPO			PFP
	Simulated Deployment Test (SDT)				
SDT 01	Shipment Test	FSPO	4.5	10	DOAP
SDT 02	Deployment Test	FSPO	4.5	10	DOAP
SDT 03	Over The Air Communications Test	FSPO	4.5	10	DOAP
SDT 04	Sunlight Test	FSPO	5.5	10	DOAP
	Day in life Test (DLT)				
DLT 01	Telemetry Retrieval Test	FSPO	1.5	4	IDP
DLT 02	Commisionning Test	FSPO	4.5	8	IDP
DLT 03	Experiment Commanding Test: Manual GS	FSPO	4.5	8	IDP
DLT 04	Data Retrieval Test: Manual GS	FSPO	1.5	4	IDP
DLT 05	Experiment Commanding Test: Automated GS	FSPO	1.5	6	MAX
DLT 06	Data Retrieval Test: Automated GS	FSPO	1.5	2	MAX
	Robustness Test (RBT)				
RBT 01	I^2C/CAN Bus inaccessible Test	FSPO	2.7	10	IDP
RBT 02	Secondary/multiple deployment system Test	FSPO	4.5	10	DOAP
RBT 03	Backdoor Telecommanding Test	FSPO	1.5	2	MFP
RBT 04	OBC Crash (No Reboot) Test	FSPO	1.5	10	MFP
RBT 05	OBC filesystem corruption Test	FSPO	1.5	10	MFP
RBT 06	Minimum Viable Mission Test	FSPO	5.5	8	MFP
RBT 07	Watchdog Test	FSPO	2.7	10	IDP
RBT 08	OBC Over The Air Re-program Test	FSPO	1.5	10	IDP

7.1 Risk Priority

To help future Cubesat missions understand the priority of each test or sub-test, we propose an associated robustness metric. This would assist Cubesat mission and project managers in understanding the level of risk they are taking on or eliminating when they choose to skip or perform a test. Repeating the tests is also highly recommended.

Following parts of FMECA methodologies (European Cooperation for Space Standardization 2009), we employ two key parameters to compute risk priority: failure likelihood and failure severity. The severity of a test is subjective and not directly quantifiable. Nevertheless, we define the following scale for the severity of failure

0 does not affect the mission

2 delays mission progression
4 isolated minor failure that does not cause cascaded consequence
6 major failure causing cascaded consequence
8 failure with irreversible cascaded consequence
10 irrecoverable failure

In contrast, the likelihood of failure is objective and can be derived from the statistics of past Cubesat missions (Swartwout 2013, 2016; Guo et al. 2014). It takes a value on a scale from 0 to 10. A failure likelihood of 10 for a specific test implies that 100% of past Cubesat failures involve one or more subsystems associated with this test.

7.2 Environmental Test

There are two major environmental tests associated with the validation process of European Cooperation for Space Standardization (2010): vibration testing and thermal vacuum testing. The type and range of the vibration test is dependent on the requirements imposed by the launch provider. The temperature range of the thermal vacuum test is determined by the designated orbit of the satellite. As an example, the QB50 mission requirement for the vacuum pressure is 10^{-5} mBar and temperature range from -20 °C to 50 °C. Since Cubesats are typically scheduled for ride-share or as secondary payloads, being unable to meet these tests on time may result in missing the designated launch slot or worse. Hence, the risk priority of this test is extremely high.

7.3 Simulated Deployment Test

Simulated Deployment Tests involve not just the simulated process of insertion into orbit, but also processes before and after the deployment process. The shipment test will involve the pre-shipment checks, non-volatile configuration of the satellite, charging or discharging the batteries to voltage levels mandated for transport, the Remove Before Flight (RBF) feature, packaging, boxing and the reverse of that process post shipment. After the deployment, the Cubesat under test needs to be able to perform GS communications over-the-air and charge its batteries via EPS and its solar panels without any external interference or telecommanding. The four subtests of SDT need to be performed in sequence and uninterrupted. It is very important that SDT be performed in Full Stack Plugs Out (FSPO) configuration to prevent a deployment failure as in the case of UNSW-EC0.

Bouwmeester et al. (2017) studied 129 satellites and found that at least 45% of all satellite failures can be allocated to electrical faults. The EPS is accountable for 27% and Command and Data Handling (CDH) for 15% of the failures encountered, which together have a major impact on reliability.

All types of electrical and mechanical faults may be exposed by Shipment, Deployment Communications and Sunlight Tests. However, more specifically, SDT 04 is not affected by the communications subsystem and the OBC. SDT 01 will verify both the hardware and software configuration process and the boxing and unboxing process, all electrical type failures. Thus, SDT 01 to SDT 03 will comprise of failures associated with the CDH and EPS and have the same likelihood of 45%. From Tafazoli (2009), we see that the solar array failures within the first year of operation has an occurrence rate of 55%.

7.4 Day in Life Test

DLTs involve executing the mission as if the Cubesat is in space in FSPO configuration. That is, executing the mission without plugging in any external electrical interfaces nor being able

to access any physical parts of the Cubesat. The satellite must run under its own power and commands, telemetry and other data, e.g., dummy payload data must be obtained using the respective radio transceiver(s).

The DLTs are mainly to test the implemented functionalities of the CDH software including limiting GS commanding and telemetry retrieval to be within the time window of the satellite in its designated orbit passing over the GS. Nominally, these tests need to be performed continually across multiple days.

These tests will ensure that documented procedures will not cause any failures, for example, an erroneous execution of ADCS or experiment payload might trigger an irreversible spin or an electrical fault that may result in permanent loss of communications or power to the rest of the satellites. Hence, DLT 02-03 will cause cascaded effect to the rest of the subsystems and has the same likelihood of occurrence as SDT 01-03. It is implied that DLT 05-06 would have been de-risked by passing tests of DLT 03-04, hence given a lower ranked severity.

7.5 Robustness Test

The Robustness Tests aim to validate one or more alternative unique methods to achieve the same critical function in the event that the primary method of achieving that critical function fails. This can sometimes refer to alternative methods of performing the function via another hardware or software that is already in place, or an intentionally implemented functional redundancy. RBT also covers implemented functionalities that serve as the last line of defence to recover against various types of failures reported in many other Cubesat missions. Issues or functionalities in RBT should never occur or be specifically relied upon in any of the subtests of SDT or DLT. Hence, the faults in all RBT sub-tests need to be manually injected. Indeed, the severity of failing the tests here would mean an irrecoverable mission failure, with few exceptions.

From Swartwout (2013), a configuration or interface failure between communications hardware has an occurrence rate of 27% which is an irreversible failure that can be prevented if repeated test of RBT 01 and RBT 07 confirms the robustness of the implemented system. The Watchdog Test (RBT 07) in the UNSW-EC0 subsystems refers to the I^2C watchdog previously described, the CSP watchdog which power cycles non-responsive GOMSpace subsystems and the ground contact watchdog which power cycles all subsystems if no uplink from GS can be detected for an excessive period of time. There are also various other forms of watchdogs. All watchdogs need to be tested for ensuring (1) the desired recovery behaviour is triggered under timeout and (2) the timeout will not occur under nominal circumstances.

RBT 03-05 and RBT 08 relate to failure prevention of CDH capabilities that are often overlooked by other system-level tests and not tested in a FSPO configuration. Given that the failure affects only CDH, Bouwmeester et al. (2017) indicates that the likelihood of this category of failure is 15%. However, if this test also prevents some form of COM or EPS failure, then the likelihood of these tests may need to be revised up to 55%. The ability of the OBC to handle such failures are paramount as an OBC crash that does not reboot or an OBC non-volatile file system corruption may mean an irrecoverable mission failure. From Bouwmeester et al. (2017), the occurrence rate of electrical faults at 45% is attributed to RBT 02 because any form of electrical fault would prevent deployment.

The Minimum Viable Missions (MVM) tests are a risk minimisation exercise. This will force the mission stakeholders to pre-define the acceptable minimum outcome of the mission in the event that full mission success is not achievable due to a single point of failure.

One form of MVM would be to have the ability to execute an experiment using only one uplink packet without handshake. This packet would arm a pre-programmed experiment sequence that will autonomously execute on the OBC. The experiment data would then be autonomously made to downlink repeatedly without any handshaking. This would render the MVM to be robust to any form of intermittent communications issue. The MVM also needs to be executable by all functions that have redundancy, subject to the resource envelope of the mission.

Over-the-air reprogramming (RBT 08) of the OBC is often an under-estimated feature. Having this capability will add mission robustness under unforeseen circumstances and may even prevent the mission from entering an irrecoverable failure state. Thus, this is given a severity ranking of 10. Furthermore, it has the added advantage of allowing the OBC to be reconfigured to pursue riskier tertiary experiments if full mission success has been achieved ahead of schedule.

During the development phase, some of the debugging commands issued directly to the OBC via its UART-GOSH were useful for operations. Instead of implementing all of those commands again so that they could be accessed nominally via its designated GS terminal program (CSPTerm), the satellite can be modified so the OBC can operated in orbit as if commands were issued from the UART-GOSH interface. This is only one example of many possible forms of backdoor telecommanding. The backdoor telecommanding test ensures that all possible alternative routes to telecommanding the OBC is available in orbit.

RBT 02 tests any alternative antenna deployment system. For example, it may be possible to issue a command to deploy a turnstile antenna via an S-band communications or the Globalstar inter-satellite link where both use patch antennas that do not require any deployment. This test can also involve deployment re-attempt sequencing, (i.e. assume that first few deployments happened under freezing temperatures), or prevention of deployment under low voltage, or an alternative hardware-only deployment logic (e.g. an EPS driven deployment).

Due to physical limitations of the overpopulated Cubesat stack, UNSW-EC0 did not have an antenna deployment sensor. Thus, there was no feedback to the OBC as to whether the command for the burn resistors had successfully released the antennas or not. A single attempt to deploy the antennas might fail if the temperature of the burn mechanism is excessively low, leading to the possibility of a partially burnt Dyneema wire. Alternatively, if the deployment were executed when the battery voltage is low, the Dyneema wire might be partially cut as the burn resistors would be unable to sufficiently heat it up. Therefore, the deployment program of the OBC firmware would periodically repeat the antenna deployment sequence until a specific non-volatile flag in the OBC file-system has been manually set by an uplink command. This is a distinctive redundancy that should be replicated in all future Cubesat missions.

7.6 Configuration Type

Partial Stack Plugs In (PSPI) and Full Stack Plugs In (FSPI) are Cubesat configurations limited only to the development phase of the Cubesat. Towards the end of the development cycle, we transitioned from unit and functional testing to testing how we were going to fly. This meant that instead of having the Cubesat plugged in for external power and serial communications link for command and debugging, the satellite should be tested under its own power with no wired physical link to it. The GS hardware and software were used to command and debug the satellite. This is the FSPO test.

An exception to the no physical link is when there are imposed lab regulations where the antennas has to be replaced with a direct coaxial line between the GS transceiver and

satellite transceiver via attenuators to provide similar signal levels to the actual flight. This meant that the GS antenna and LNA along with the satellite antenna could not be a part of the "test like you fly" tests—all of these components are usually tested as units either by the vendor in the case of the satellite antenna or by the GS team. In the development of UNSW-EC0, the GS equipment was used to communicate and operate the satellite from the GS terminal for the FSPO tests. However, this was not done throughout the development cycle due to the increased time required by this process and the limited debugging capability compared to the physical interfaces to the satellite.

8 Conclusion

This paper presented the technical lessons learned and results achieved through a six year development effort that started in 2012 and culminated in the launch, recovery and operation of the UNSW-EC0 Cubesat until its re-entry on 4th December 2018. It also presented various in-orbit results of the CDH, ADCS and payload experiments that verified the nominal operations of UNSW-EC0.

UNSW-EC0 was a 2U Cal Poly compliant Cubesat that formed UNSW Sydney's contribution to the QB50 project. The satellite was built on a very small budget and its development was constrained by limited resources and relied on a team of volunteer staff and students. Importantly, this was UNSW's first ever satellite, meaning that the project was valuable as a capability building exercise.

Despite the limited budget and resources, UNSW-EC0 was successfully launched and operated. After a false start where a communication link could not be established with the satellite, the UNSW Sydney team conducted a thorough and effective failure mode analysis and UNSW-EC0 was recovered. Analysis of the problem led to the hypothesis that it was due to the satellite being unable to deploy its stowed antennas and the reduced antenna gain did not permit the communications link to be closed. A recovery plan was then devised and UNSW-EC0 was able to deploy its antennas and continued to perform nominally for the remaining duration of its designed mission lifetime.

UNSW-EC0 has delivered a number of important lessons on resource management and engineering issues in low-cost Cubesat science projects. Both the failure recovery described here and the operational procedures that were put in place for the remainder of the mission were described. Our experience showed that a combination of existing and new methods of Cubesat testing and a system-level validation approach were crucial to overcoming the various in-orbit issues that were encountered. It also revealed that a Full Stack Plugs Out (FSPO) test is crucial to prevent any non-equivalence to the in-orbit scenario. It is hoped that these lessons prove useful to future Cubesat-scale space science missions, especially for teams that are resource constrained.

Acknowledgements We would like to thank the UNSW BLUESat student society for their help in setting up and maintaining the UNSW GS, Thai Loi from UNSW Legal for providing legal support, and the UNSW Faculty of Engineering and UNSW Central for providing funding support.

We would like to thank the I-INSPIRE II team at University of Sydney, in particular Jiro Funamoto and Iver Cairns, the BLUEWren team at Australia National University Canberra including Christine Charles and Dmitris Tsifakis for their spirited collaboration and mutual assistance.

We would also like to thank members of the amateur radio community, Jan Van Muljwijk (PA3FXB) from the CAMRAS team who operated the Dwingeloo 25 m Radio Telescope in Netherlands, Daniel Estévez (EA4GPZ) who implemented the open source radio beacon decoder Jan Van Gills (PE0SAT), Mike Ruppprecht (DK3WN), and many other amateur radio operators around the world who contributed to the beacon collection.

Special thanks to team members of Von Karman Institute and ISIS Space for being accommodating during the satellite launch integration process.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

E. Aboutanios, J.W. Cheong, The UNSW-EC0 mission: overview of the recovery, commissioning and mission plan, in *European Cubesat Symposium 2017* (2017), p. 46

L.K. Alminde, J. Christiansen, K.K. Laursen, A. Midtgaard, M. Bisgaard, M. Jensen, B. Gosvig, A.A. Birklykke, P. Koch, Y. Le Moullec, M. Bisgard, M. Jensen, B. Gosvig, A.A. Birklykke, P. Koch, Y.L. Moullec, GomX-1: a nano-satellite mission to demonstrate improved situational awareness for air traffic control, in *26th Annual AIAA/USU Conference on Small Satellites* (2012)

L.W. Blum, Q. Schiller, X. Li, R. Millan, A. Halford, L. Woodger, New conjunctive CubeSat and balloon measurements to quantify rapid energetic electron precipitation. Geophys. Res. Lett. **40**(22), 5833–5837 (2013)

J. Bouwmeester, M. Langer, E. Gill, Survey on the implementation and reliability of CubeSat electrical bus interfaces. CEAS Space J. **9**(2), 163–173 (2017)

M. Brodie, J.W. Cheong, B.J. Southwell, Implementation of collaborative and automated GSs for UNSW-EC0 and INSPIRE2, in *17th Australian Space Research Conference* (2017)

E. Cetin, O. Diessel, T. Li, J.A. Ambrose, T. Fisk, S. Parameswaran, A.G. Dempster, Overview and investigation of SEU detection and recovery approaches for FPGA-based heterogeneous systems, in *FPGAs and Parallel Architectures for Aerospace Applications* (Springer, Berlin, 2016), pp. 33–46

J.W. Cheong, B. Southwell, C. Lam, J. Bultitude, W. Andrew, S. Green, B. Osborne, A.G. Dempster, E. Aboutanios, W. Crowe, Design and Development of the UNSW QB50 Cubesat—EC0, in *International Astronautical Congress* (2016)

J.W. Cheong, B. Southwell, E. Glennon, A.G. Dempster, E. Aboutanios, Progress and update of UNSW EC0: Australia's first CubeSat trio in orbit, in *18th Australian Space Research Conference* (2018), p. 78

M. Choudhury, E. Glennon, A.G. Dempster, P. Mumford, Characterization of the Namuru V3.2 spaceborne GPS receiver, in *12th Australian Space Science Conference*, vol. 55 (2012)

Data61, The seL4 microkernel (2018). https://ts.data61.csiro.au/projects/seL4/. Accessed: 2018-Dec-12

M.R. Drinkwater, R. Haagmans, D. Muzi, A. Popescu, R. Floberghagen, M. Kern, M. Fehringer, The GOCE gravity mission: ESA's first core Earth explorer, in *Proceedings of the 3rd International GOCE User Workshop* (2006)

A. Elbers, *The Rise of Radio Astronomy in the Netherlands*, 1st edn. (Springer, Berlin, 2017). https://doi.org/10.1007/978-3-319-49079-3

European Cooperation for Space Standardization, *Failure Modes, Effects (and Criticality) Analysis (FMEA/FMECA)* (Standard, Noordwijk, 2009)

European Cooperation for Space Standardization, *Space Engineering—Verification Guidelines* (Standard, Noordwijk, 2010)

E.P. Glennon, A.G. Dempster, Improving sensitivity on Kea CubeSat GPS receivers, in *International Global Navigation Satellite Systems Society, IGNSS Symposium* (2016)

E. Glennon, K. Parkinson, P. Mumford, N. Shivaramaiah, Y. Li, R. Li, Y. Jiao, A GPS receiver designed for CubeSat operations, in *Australian Space Science Conference* (2011), pp. 26–29

E. Glennon, J. Gauthier, M. Choudhury, A. Dempster, K. Parkinson, Synchronization and syntonization of formation flying CubeSats using the Namuru V3.2 spaceborne GPS receiver, in *Proceedings of the ION 2013 Pacific PNT Meeting*, Honolulu, HI, USA (2013), pp. 23–25

E.P. Glennon, J.P. Gauthier, M. Choudhury, K. Parkinson, A.G. Dempster, Project Biarri and the Namuru V3.2 Spaceborne GPS Receiver, in *International Global Navigation Satellite Systems Society, IGNSS Symposium* (2013)

GomSpace, Cubesat space protocol (CSP): network-layer delivery protocol for CubeSats and embedded systems (2011). https://bytebucket.org/bbruner0/albertasat-on-board-computer/wiki/1.%20Resources/1.1.%20DataSheets/CSP/GS-CSP-1.1.pdf?rev=316ebd49bed49fdbb1d74efdeab74430e7cc726a. Accessed: 2018-Dec-12

J. Guo, L. Monas, E. Gill, Statistical analysis and modelling of small satellite reliability. Acta Astronaut. **98**, 97–110 (2014)

H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, R. Twiggs, Cubesat: a new generation of picosatellite for education and industry low-cost space experimentation, in *Small Satellite Conference*, AIAA (2000)

Joint Functional Component Command for Space, Space-track (2018). https://www.space-track.org. Accessed: 2018-Dec-12

M. Langer, J. Bouwmeester, Reliability of CubeSats—statistical data, developers' beliefs and the way forward, in *Proceedings of 30th Annual AIAA/USU Conference on Small Satellites* (2016)
W.J. Larson, J.R. Wertz, Space mission analysis and design. Tech. rep., Torrance, CA (United States). Microcosm, Inc. (1992)
J. Muylaert, R. Reinhard, C. Asma, J. Buchlin, P. Rambaud, M. Vetrano, QB50: an international network of 50 CubeSats for multi-point, in-situ measurements in the lower thermosphere and for re-entry research, in *ESA Atmospheric Science Conference*, Barcelona, Spain (2009), pp. 7–11
National Academies of Sciences, Engineering, and Medicine, *Achieving Science with CubeSats: Thinking Inside the Box* (The National Academies Press, Washington, DC, 2016). https://doi.org/10.17226/23503. https://www.nap.edu/catalog/23503/achieving-science-with-cubesats-thinking-inside-the-box
NXP Semiconductors, *I2C-Bus Specification and User Manual* (Standard, Netherlands, 2014)
B. Osborne, E. Aboutanios, A. Dempster, E. Cetin, G. Heiser, E. Glennon, UNSW EC0 CubeSat design: experiments in radiation tolerance critical systems, GNSS remote observation and 3-D printed satellite structures, in *5th European Cubesat Symposium* (2013), p. 41
K. Parkinson, P. Mumford, E. Glennon, N. Shivaramaiah, A. Dempster, C. Rizos, A low cost Namuru v3 receiver for spacecraft operations, in *International Global Navigation Satellite Systems Society, IGNSS Symposium* (2011), pp. 15–17
R. Plauché, Building modern cross-platform flight software for small satellites, in *Small Satellite Conference 2017* (2017)
A. Poghosyan, A. Golkar, Cubesat evolution: analyzing CubeSat capabilities for conducting science missions. Prog. Aerosp. Sci. **88**, 59–83 (2017)
W.W. Royce, Managing the development of large software systems: concepts and techniques, in *Proceedings of the 9th International Conference on Software Engineering* (IEEE Computer Society Press, Los Alamitos, 1987), pp. 328–338
SatNOGS, SatNOGS DB (2018). https://db.satnogs.org/satellite/42723/. Accessed: 2018-Dec-12
A. Scholz, T.H. Hsiao, J.N. Juang, C. Cherciu, Open source implementation of ECSS CAN bus protocol for CubeSats. Adv. Space Res. **62**(12), 3438–3448 (2018)
A. Scholz, J.N. Juang, P. Mader, J. Schlegel, M. Starcik, Spacecan—a low-cost, reliable and robust control and monitoring bus for small satellites. Acta Astronaut. **161**, 1–11 (2019)
D. Selva, D. Krejci, A survey and assessment of the capabilities of Cubesats for Earth observation. Acta Astronaut. **74**, 50–68 (2012)
B.J. Southwell, The UNSW-EC0 GS, in *CubeSat Innovation Workshop* (UNSW, Sydney, 2018)
B.J. Southwell, J.W. Cheong, UNSW-EC0 CubeSat in orbit: challenges and results, in *17th Australian Space Research Conference* (2017)
M. Swartwout, The first one hundred CubeSats: a statistical look. J. Small Satell. **2**, 213–233 (2013)
M. Swartwout, Secondary spacecraft in 2016: why some succeed (and too many do not), in *IEEE Aerospace Conference Proceedings* (2016). https://doi.org/10.1109/AERO.2016.7500791
M. Tafazoli, A study of on-orbit spacecraft failures. Acta Astronaut. **64**(2–3), 195–205 (2009)
A. Toorian, K. Diaz, S. Lee, The CubeSat approach to space access, in *Aerospace Conference, 2008 IEEE* (IEEE, Pasadena, 2008), pp. 1–14
S. Waydo, D. Henry, M. Campbell, Cubesat design for LEO-based Earth science missions, in *Proceedings, IEEE Aerospace Conference*, vol. 1 (2002), p. 1
K. Woellert, P. Ehrenfreund, A.J. Ricco, H. Hertzfeld, Cubesats: cost-effective science and technology platforms for emerging and developing nations. Adv. Space Res. **47**(4), 663–684 (2011)
H. Zimmermann, OSI reference model—the ISO model of architecture for open systems interconnection. IEEE Trans. Commun. **28**(4), 425–432 (1980)