Brandon Lucia, Brad Denby, Zachary Manchester, Harsh Desai and Emily Ruppel *Carnegie Mellon University, Pittsburgh, PA*
Alexei Colin *USC Information Sciences Institute, Arlington, VA*

Editor: Shadi Noghabi

COMPUTATIONAL NANOSATELLITE CONSTELLATIONS: Opportunities and Challenges

As rocket launch cadences increase, access to space rises dramatically – setting the stage for the next space industry surge. New, smaller, and less expensive satellites – now "nanosatellites" – can be deployed en masse to form constellations of hundreds, thousands, or even tens of thousands of devices [27, 40, 41, 16, 17, 18, 43]. A constellation of nanosatellites equipped with sensors (e.g., visual or hyperspectral cameras, particle detectors, or magnetometers) and radios provides a first-time opportunity for orbital swarm sensing to synthesize data from the unique vantage point of low-Earth orbit (LEO).

Applications of this data are diverse and hold promise to benefit the societies of the globe. Sensor-equipped nanosatellites could monitor agriculture for signs of flooding, crop pests, or nutrient deficiencies [46, 19]. Tiny satellites could track wildfires and the pollutants they produce with high temporal resolution, identifying hotspots, outbreaks, and imperiled residential areas for first responders. As tens of thousands of dust-like satellites reenter the atmosphere in order to avoid space debris, they could take measurements and provide unprecedented, fine-grained climatological data before vaporizing – laying the groundwork for a similar deployment above Venus to study its hostile atmosphere. Constellations of capable space sensors are limited only by the imagination of designers and the needs of Earth dwellers.

However, these promises remain elusive because today's mission operations rely on humans-in-the-loop. In most satellite missions, a human operator manually sends commands to each satellite individually and awaits replies from each satellite individually. One command might instruct a satellite to focus its camera at a particular point on Earth; another might activate a satellite's reaction wheels to rotate a radio toward a particular ground station and downlink collected wildfire images; yet another might activate a micro-ion thruster [26] to change its orbit characteristics. Dependence on individual, fine-grained human-to-satellite commands impedes effective operation of emerging nanosatellite capabilities. As constellation population increases, this system architecture becomes infeasible. Human-operated command and control uplinks are a critical bottleneck to new, sophisticated constellation operations and applications.

The physical design of nanosatellites and ground infrastructure also limit constellation capabilities and applications. With only a small battery, or no battery at all [8, 7, 6], the activity of a tiny nanosatellite is limited by the energy it collects. Ultra-limited-energy operation requires ultra-low-power sensors, onboard computers, radios, and actuators. An energy-constrained radio has limited uplink and downlink datarates. In most cases we have studied, these datarate limits

Illustration, istockphoto.com

mean that the total amount of, e.g., sensor readings that can be downlinked, are only a small fraction of the total readings that can be collected [8]. Ground infrastructure also has a cost. Satellites in polar orbits benefit most from radio ground stations deployed near the poles, usually at high installation and operation costs. In a large constellation, ground equipment may be oversubscribed at some times (e.g., when many satellites are near the same ground equipment simultaneously) and undersubscribed at other times (e.g., when no satellites are in range).

The sidebar, "Motivation for, and Benefits of, Orbital Edge Computing," provides an overview of the benefits of edge computing in space, the communication bottleneck faced by today's nanosatellites, and trends toward smaller, cheaper satellites.

This paper describes a new approach to nanosatellite design and operation, which we call "orbital edge computing." Orbital edge computing (OEC) eliminates the humans-in-the-loop, making each tiny satellite a capable, autonomous system, and transforming a constellation of nanosatellites into a sophisticated, orbital sensing and data processing infrastructure. Operating autonomously without human interaction, each nanosatellite in a constellation collects and processes all sensor data using, e.g., machine learning and inference algorithms without the high costs associated with downlinking all data to Earth. When data processing identifies signals or features of interest, the satellite can transmit only that interesting data to Earth, alleviating datarate requirements, reducing ground infrastructure costs, and increasing the "signal to noise ratio" of sensor data received on Earth.

This paper provides an introduction to the design space of orbital edge computing systems, including key opportunities and challenges. We focus on computer systems and computing challenges; the addition of sophisticated computing and data processing to nanosatellites is the key novelty of orbital edge computing. This design space spans a spectrum of nanosatellite size, capability, and constraints — from CubeSat [33]-scale devices to vanishingly tiny chipsats [31, 47]. Here, we describe two concrete designs: a recently launched, proof-of-concept computational chip-scale nanosatellite, and a new, flexible, batteryless computational nanosatellite bus that provides high mission capability at extremely low cost for future, large-scale constellations. This paper is intended as a call to action for computer system researchers to reach out to the space computing systems community in order to find and solve interdisciplinary, cross-cutting orbital edge computing research problems.

ORBITAL EDGE COMPUTING IN NANOSATELLITES

Orbital edge computing shifts data processing responsibilities from Earth to nanosatellite constellations. This shift avoids reliance on downlink availability and can eliminate the need for manual operation. Supporting orbital edge computing under the unique constraints of nanosatellite form factors raises new challenges.

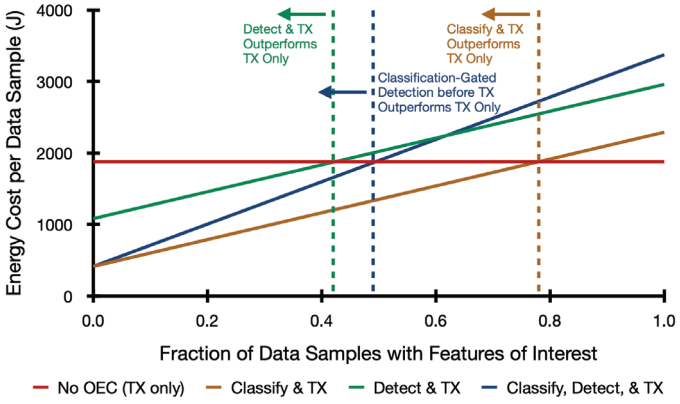
Challenges. Minimizing the size, weight, power, and cost (SWaP-C) of individual nanosatellites supports increased constellation device counts. However, low SWaP-C limits complexity and results in small solar panel surface areas that provide little power. The small size of nanosatellites provides little — or no — volume for energy storage. Thus, the tiniest devices operate *intermittently*, i.e., only when energy is available [28, 6, 47]. Orbital edge computing in such extreme power conditions necessitates extreme energy efficiency for data processing with machine learning, control and motion planning, and communication. When available, actuation (e.g., attitude control) is often dimensionally limited (i.e., underactuated) and almost always imposes a high energy cost both for planning and executing actuation.

Research Opportunities. In the context of such challenges involved in developing orbital edge computing for computational nanosatellites, computer systems research opportunities abound — especially in power and energy management, intermittent computing techniques, and computer architecture and hardware specialization.

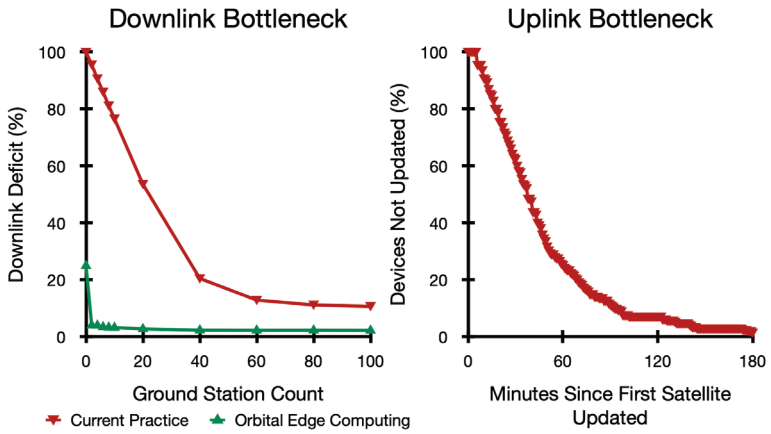
Very small satellites depend on power systems that carefully manage small quantities of harvested energy. Solar panels, which are the main method of energy collection for nanosatellites, are necessarily very small. Absent complex and operationally risky deployable solar panel arrays, nanosatellites have relatively low input power. A 1p pocket-qube [37] with surface-mounted panels may harvest at most 500 mW of power; a chipsat may harvest just 1–10 mW of power.

These power levels are too low to support continuous operation of all but the most energy-efficient computing hardware — a key research opportunity. Energy-efficient computing using highly capable, ultra-low-power (i.e., 100 µW–1 mW) computer architectures is essential, especially for the most SWaP-C constrained nanosatellites. Some designs may include mobile GPUs [35] or hardware specialized for neural network inference and computer vision [4, 10, 15]. However, owing to a wide breadth of still-emerging applications and a need for near-ASIC efficiency, nanosatellites should include programmable accelerator hardware, leveraging dataflow [36, 34], vector execution [25], or combinations of these techniques [14]. Computational nanosatellites may also leverage design methodologies that fit specialized architectures to power, workload, and latency constraints [9].

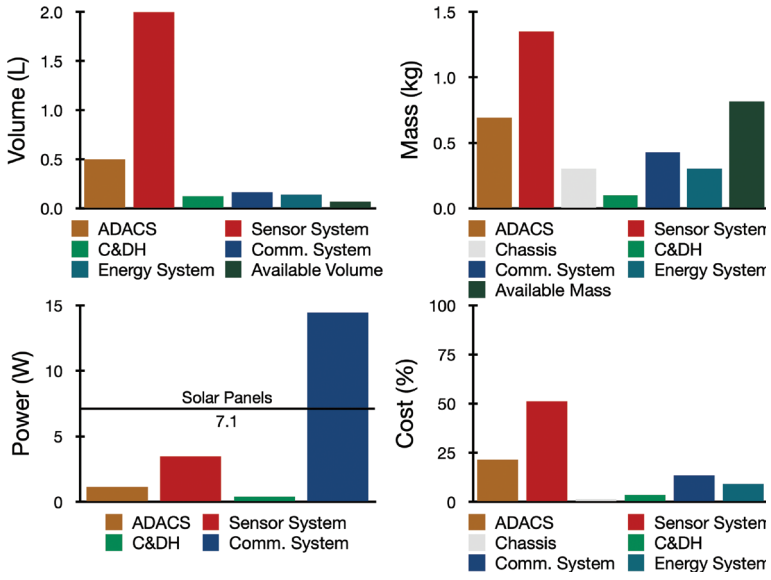
(a)



(b)



(c)



Motivation for, and Benefits of, Orbital Edge Computing

(a) Even with sufficient radio ground station infrastructure to downlink all data, the energy cost of radio transmission is high. Onboard, neural network inference selects only data samples with features of interest to reduce data volume and energy consumption. In some cases, less than five percent of raw data contain features of interest. We evaluate the energy benefits of CNN classification and object detection to triage raw data before transmission. Classification requires less power, while detection drastically decreases data quantity. Composing the two into classification-gated detection provides some of both benefits.

(b) Downlinks: With each revolution, satellites in existing constellations can collect more data than can be downlinked — resulting in a "downlink deficit" that compounds over time. This deficit is eliminated with OEC. Uplinks: Even a small command or configuration update can take hours to be sent up to all satellites in existing constellations. Autonomy with OEC can eliminate the need for these uplinks.

(c) Nanosatellites have new size, weight, power, and cost constraints. Small payload volumes and tiny solar panel surface areas place limitations on radio communication. OEC can address these limitations using new, low-power, onboard computers.

A power-adaptive microarchitecture model [9], enables a nanosatellite to modulate its efficiency and performance so as to best use scarcely available energy as input power varies. With high input power, a system charges quickly; the latency of a computing task is determined by the performance of the computer. With low input power, a system charges slowly; the time to charge dominates the latency to perform a computing task. Adaptive architectures are the key to minimize latency. At extreme high power, an architecture should compute as fast as possible regardless of energy efficiency. At low power, an architecture should compute as efficiently as possible regardless of performance. Between these extremes lies a tradeoff space of performance and efficiency for a given input power. Figure 1 shows how both energy collection time (i.e., input power) and computing time determine which architecture has the best end-to-end latency for a computing task (in this case, matrix multiplication). The data are from our PHASE architectural model [9] and show, at each input power level, whether it is best to compute using the high-performance/ low-efficiency ARM A15 (big), the performance/ efficiency-balanced ARM A7 (LITTLE), or the ultra-low-energy MANIC vector-dataflow processor [14]. As input power varies, the processor with lowest end-to-end latency varies; MANIC for the lowest input power (P_{input} < 10 mW), LITTLE for medium input power (10 mW ≤ P_{input} < 700 mW), and big for the highest input power (P_{input} ≥ 700 mW). A nanosatellite must be able to select the best computer architecture option as its harvested power varies due to unpredictable tumbling, actuation-induced movement, and regular orbital power variation (i.e., moving into Earth's umbra). Future nanosatellites should include such power-adaptive computing components.

A configurable power system gives a nanosatellite the ability to finely manage its energy and adapt to varying power consumption and input power supply. To perform any computational work, a system must collect and buffer a useful quantum of harvested energy to later use for computing. Batteries are one option for storing large quantities of energy but, being large and heavy, they are often a mismatch for the SWaP-C requirements of a nanosatellite. Batteries are poorly adapted to the harsh temperatures of space [7] and suffer short lifetimes when repeatedly charged and discharged.

A research opportunity lies in designing a computational nanosatellite without batteries. Promising alternatives to batteries are supercapacitors and ceramic capacitors, both with unlimited lifetimes and environmental robustness but with a comparatively lower energy density [6]. Supercapacitors are an order of magnitude more energy dense than ceramic capacitors, which proportionally reduces the size of the satellite and lowers the voltage to which the capacitor must be charged, increasing charging efficiency. This advantage in energy density comes at a cost of a limit on the load current. Load current causes a temporary voltage drop across the supercapacitor's equivalent series resistance (ESR) that decreases the voltage available to the load. A voltage drop below the load's minimum operating voltage renders stored energy inaccessible. Unmitigated, such a drop may cause surprising power failures. A research opportunity exists in designing *load-aware* power systems and in building software and hardware to schedule computations in order to avoid critical voltage drops. Load-aware code scheduling moves high current operations earlier in the supercapacitor's discharge cycle, leaving additional voltage budget for a drop over the ESR. Load-aware power-systems reconfigure the capacitor bank [6] to decrease the ESR when high-current tasks are scheduled. Both hardware and software solutions are promising opportunities to increase the capabilities of computational nanosatellites without increasing the volume devoted to energy storage.

Intermittent computing [28] is an approach to batteryless system design that allows a system to tolerate highly variable energy availability and periodic unavailability. An intermittent nanosatellite computes when energy is available, resulting in bursts of activity interspersed with periods of inactivity and energy collection. Intermittent computing is maturing for Earth-bound, ultralow-power smart edge systems [28, 1, 29, 45, 30, 39, 24]. Intermittent computer systems are progressing to support sophisticated machine learning and inference [13, 14, 38] — capabilities important to sensor-driven nanosatellite missions. Workloads with significant onboard processing of sensed data may require more energy than is available in a single burst of activity, which are determined by the capacity of the energy storage. For example, reading a sensor, constructing a packet, and transmitting the packet may each need to happen in a different activation burst, and a productive system would offer support for such decomposition leveraging non-volatile memory. Reliable software and hardware operation is a key mission requirement for nanosatellites, especially if operating intermittently. Formal behavioral models of intermittent software execution [42, 2] pave the way for provably correct and secure nanosatellite systems, though much of this area remains unexplored.

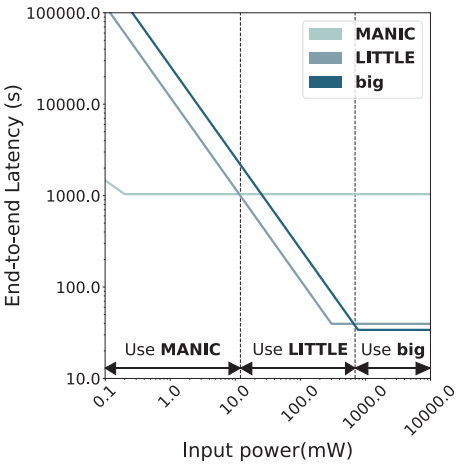


FIGURE 1. The end-to-end latencies of a matrix-multiplication workload on three different processors — the ARM A7 (LITTLE), the ARM A15 (big), and the MANIC vector processor — across input power levels. End-to-end latency includes the time to execute the workload and the time to collect the energy required to run the workload. As input power varies, a different processor provides the lowest end-to-end latency. Future nanosatellites should switch between energy-efficient and high-performance modes depending on the input power level to maximize end-to-end performance.

FROM COMPUTATIONAL NANOSATELLITE TO COMPUTATIONAL CONSTELLATION

A key benefit of small, cheap nanosatellites is the ability to launch them in large constellations. A constellation of computational nanosatellites creates new research challenges and opportunities for distributed sensing, computing and communication.

Challenges. In a constellation, devices must coordinate to distribute sensing and computing tasks, potentially for multiple application workloads, co-resident on the constellation. Satellites execute actuation, control, and motion planning tasks to change their attitude; attitude changes may point a sensor at a target, or point a radio antenna at a peer satellite (for crosslinking) or a radio ground station (for downlinking). Devices must continuously geolocate themselves in order to trigger geolocalized actions, e.g., sensing or downlinking while passing over a particular ground station on Earth. A workload's sensing and computing tasks should be distributed and coordinated across satellites in a constellation with a minimum of reliance on low-data-rate crosslinks and unreliable downlinks.

Research Opportunities. The research challenges posed by nanosatellite constellations create new computer systems research opportunities.

Distributed orbital edge computing requires new software and hardware for constellation-scale multi-tenancy of complex machine learning workloads, each competing for energy, sensing and actuation opportunities, sensor data, and link bandwidth. Constellations need new operating systems and schedulers that manage this complex web of constraints to meet application requirements. A scheduler might distribute processing of a single, large sensor input across many satellites in a pipeline (i.e., tiling an 8K hyperspectral image) [8, 7]. Coordinating distributed computing explicitly with optical or radio cross-links is difficult, but promising even in small cubesats. Coordinating distributed computing based solely on geolocation signals — triggering each satellite's image capture and subset of processing based on a static GPS-guided policy — obviates the need for cross-links, at some cost in workload flexibility. Leveraging distributed constellation-scale computing for continuous, on-orbit machine learning training (e.g., federated learning [20] to detect new objects of interest in Earth images) without human involvement is an appealing, though challenging, future use for nanosatellite constellations.

Actuation is key to reliable operation of almost all satellite remote sensing systems. Cameras must be pointed at targets, antennas must be pointed at ground stations, solar panels must be pointed at the sun, and station-keeping must be performed to maintain desired orbit characteristics. As satellites become smaller, the SWaP-C associated with traditional actuators like reaction wheels and thrusters becomes prohibitive. One strategy for reducing actuator SWaP-C on small satellites is to take greater advantage of the LEO environment. Control torques for pointing can be generated by interacting with the Earth's magnetic field using electromagnets (commonly known as magnetorquers) [12], or by shaping the spacecraft's mass distribution to take advantage of the gravity-gradient effect [32]. Similarly, control forces for orbital station-keeping can be produced by altering satellites' cross-sectional area to modulate their drag [11]. System designers must design computational nanosatellites to consider the energy and time cost of these actuations. In a resource-constrained system, the designer must also consider the effect of imperfect actuation on application output quality; e.g., with insufficient stored energy to point a camera, will an off-axis image be "good enough" for object recognition?

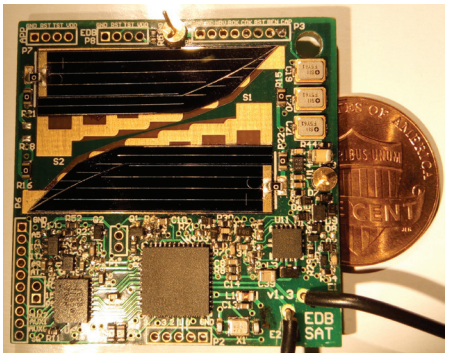


FIGURE 2. EDBsat chipsat (front). MCU and radio in the center, 3 of 10 supercapacitors in the top right, pads for attaching the V-dipole antenna in the bottom right; sensors on the back side (not visible).

Control and motion planning become significantly more challenging on small, resource-constrained spacecraft. While novel actuation methods can consume significantly less mass, volume, and power, they also suffer from underactuation — that is, they cannot instantaneously produce forces or torques in all directions. As a result, control algorithms must explicitly reason about not only power, energy, temperature, and torque constraints, but also the time-varying nature of the LEO environment and complex physical interactions with the spacecraft over long planning horizons. Developing optimization or machine-learning based algorithms that can address these control challenges within the available computing resources onboard a nanosatellite is a major research challenge.

NANOSATELLITE DESIGN AND DEPLOYMENT CASE STUDIES

EDBsat: design and deployment

Illustrating the research challenges of computational nanosatellite constellations, we executed a technology demonstration mission launching two extreme, low-SWaP-C chipsats into LEO as part of the larger, KickSat-2 chipsat launch mission. Our chipsats implemented a custom "EDBsat" design on standalone 35x35x4 mm printed circuit boards (PCBs) deployed from the KickSat-2 carrier cubesat. EDBsat's ultimately successful mission was to study the efficacy of intermittent computing in nanosatellites [5]. The system's primary design objective was to maximize the likelihood of receiving downlinked data despite unreliable power, sporadic communication opportunities, a low data-rate link, and a processor and memory not hardened to the effects of radiation.

Solar panels occupy 50% of the four-layer, double-sided PCB's surface area — providing unreliable power that varies with the orientation of the board relative to sunlight. EDBsat lacks attitude control and tumbles unpredictably after deployment. EDBsat's design includes a 66 mW radio transmitter, which could operate directly from the two space-grade solar panels (by TriSolX) — theoretically capable of delivering 96 mW (assuming an ideal orientation towards the sun and a light intensity of 1322 W/m² in LEO [21]). This narrow design margin risks catastrophic mission failure if the device never approaches the ideal orientation. Instead of this risky power system, EDBsat uses a custom circuit to boost panel voltage and accumulate energy into an array of low-profile supercapacitors that release energy once charged. The circuit[1] is built around

[1] https://github.com/CMUAbstract/releases/blob/master/EDBsat.md

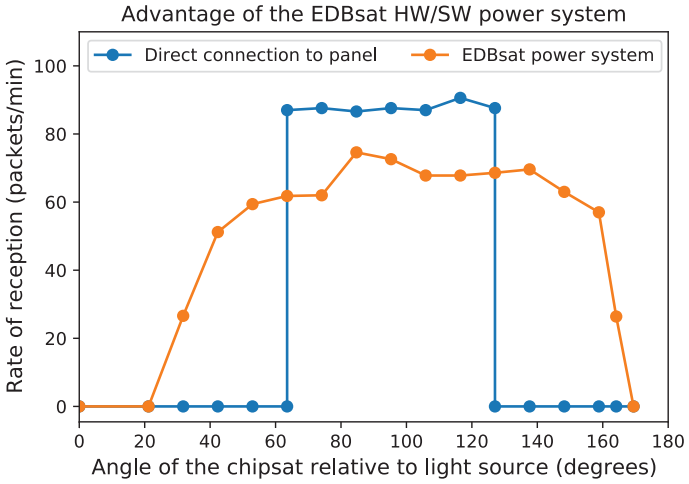


FIGURE 3. The envelope of acceptable orientation of the EDBsat chipsat for mission success, with the proposed HW/SW power system design compared to a baseline design with the load connected directly to the solar panels. The baseline delivers more energy to the load, since it avoids lossy conversion. However, as the chipsat rotates and input power drops even slightly, the baseline is unable to deliver any of it to the load at the required voltage of at least 2 V. Light was sourced from a halogen lamp in a Stocker-Yale ImageLite 20 through optical guides to a distance of 5 cm from the chipsat, at the unit's maximum intensity, at which the panels output 96 mW at 2 V as measured by a Keysight B2902A source-measure unit. Each datapoint was collected for 5 minutes.

a BQ25504 boost IC with a custom bypass path that avoids the booster's inefficient, low-voltage "cold start" mode. A load-side boost IC maximizes the usable energy available in the capacitor, trading efficiency for a smaller capacitor volume. As the experiment in Figure 3 demonstrates, the power system's main value is its assurance that the payload will activate even if the weak input power never exceeds 15 mW (7 mA @ 2.1V) – an 84% safety margin for LEO missions and a start for deep space chipsat missions.

EDBsat's downlink is weak, available sporadically (only when passing over a receiver), and — owing to power constraints — transmitting at just 13 dBm over < 160 km. EDBsat uses KickSat's custom spread spectrum implementation, which incurs high redundancy per bit and supports 125 bps datarate. The redundancy maximizes the likelihood of receiving and decoding data even using a cheap receiver and despite loss and interference from collisions. The encoder and MSK modulator run on EDBsat's commodity microcontroller (MCU) that includes an ISM band radio IC. Decoding on the ground station receiver uses SDR. To further minimize receiver cost, we built highly optimized software that runs on a multi-core, ARM-based, single-board computer (SBC) to parallelize decoding using OpenMP. The receiver includes a rechargeable battery (for portability), power conditioning circuits, an amplifier, the SBC, and a low-cost yagi antenna. The total cost of the receiver is around $150.

EDBsat operates intermittently — only as the power system stores sufficient operating energy — and its commodity MCU is not hardened to radiation-induced failures. To tolerate hard errors or errors in non-volatile memory due to radiation damage or insufficient energy for a Flash erase, we deployed two independent software payloads. The first "beacon" payload constructs and transmits a single identifier packet using at most the amount of energy stored in EDBsat's capacitors. The second "sense and log" payload reads a magnetometer and IMU and logs data in a sequence of four non-volatile *time series compression buffers*.

Each buffer, upon filling, averages its values and logs the averages into the next buffer in the sequence. These buffers encode time series data concisely. Collecting, encoding, and logging data may use more energy than EDBsat's capacitors can store at once; execution proceeds intermittently.

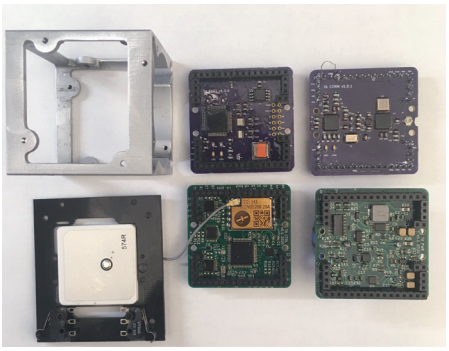


FIGURE 4. TA-1 circuit boards, aluminum chassis and FR4 base plate.

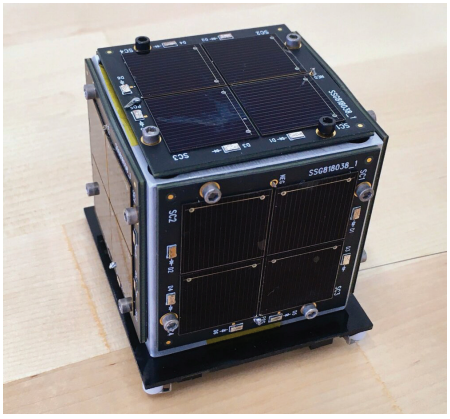


FIGURE 5. The TA-1 PocketQube nanosatellite.

For correct intermittent operation, EDBsat uses the Chain task-based intermittent programming framework [5]. EDBsat separates the two payloads further, giving each dedicated capacitor banks and power system circuits to ensure that a catastrophic failure of the sensing payload does not jeopardize the beacon application.

After EDBsat's deployment, we tracked the KickSat-2 cubesat's six-minute daily pass over our ground stations, logging IQ data. We validated the mission's success, by successfully decoding beacon payload packets collected by multiple different receivers. The EDBsat mission shows how SWaP-C constraints creates new challenge for an otherwise simple LEO mission, highlighting the need for new system designs adapted to these challenges.

Tartan-Artibeus-1

With dimensions of 5x5x5cm, PocketQubes reduce satellite cost further than traditional nanosatellites [33, 3] while providing more

power and volume than chipsats. Tartan-Artibeus-1 (TA-1) is an open-source PocketQube support system design for flexible workloads, multi-tenancy, and batteryless operation. TA-1 includes power and control subsystems that perform energy harvesting and power conditioning as well as basic flight systems tasks. The TA-1 bus exposes power rails, control signals and communication lines to allow additional subsystem PCBs to be stacked directly on top of the power and control systems.

Similar to EDBsat, TA-1's power subsystem uses dual buck/boost converters to harvest solar energy into a super-capacitor and to provide a stable output voltage. TA-1 seeks to enable the largest *usable* energy capacity from off-the-shelf super-capacitors that fit the PocketQube form factor, so the power system is designed to boost the capacitor voltage to 5.5 V and extract down to 2.0 V. The power subsystem uses current isolation switches to provide separate voltage rails for stacked subsystems and provides (dis)enable signals for each switch. Finally, the power subsystem includes an I2C enabled ADC that monitors the harvested energy and the super-capacitor's charge level.

The control subsystem uses a low-power MCU with byte addressable, non-volatile memory [44] to minimize the energy cost of basic control software and checkpointing [1, 30]. The TA-1 bus design does not allow communication between subsystems on separate power rails unless it is facilitated by the control MCU. The control subsystem includes level shifters to tolerate voltage drops on the subsystem power rails while maintaining communication to the subsystems.

We demonstrate TA-1's extensibility by adding two application subsystems, and equipping the power system with a 5.6 F super-capacitor [22]. The first application is a radio subsystem based on the OpenLST Open Radio Solution [23]. The TA-1 power system supports bursts of operation from the radio subsystem, even if no energy is harvested concurrently. The radio subsystem draws an average of 100 mA and peaks at 180 mA when sending and receiving wireless transmissions in the ISM band. To demonstrate the TA-1 bus, we built an application that receives and stores a wireless transmission on the control subsystem, then appends the received message to power system data and transmits the new packet. The second application subsystem is an accelerator that includes a Cortex-M4 for more performant processing than the control MCU can provide. The acceleration subsystem has a bootloader that allows the control subsystem to update the accelerator's software over UART. We demonstrated that an arbitrary program can be stored in the control MCU's memory and passed to the acceleration subsystem.

REFERENCES

[1] D. Balsamo, A.S. Weddell, A. Das, A.R. Arreola, D. Brunelli, B.M. Al-Hashimi, G.V. Merrett, and L. Benini. 2016. Hibernus++: A self-calibrating and adaptive system for transiently-powered embedded devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 35(12):1968-1980.

[2] G. Berthou, P.-E. Dagand, D. Demange, R. Oudin, and T. Risset. June 2020. Intermittent computing with peripherals, formally verified. In *LCTES '20 - 21st ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, 85–96, London/Virtual, United Kingdom.

[3] J. Bouwmeester, S. Radu, M. Uludag, N. Chronas, S. Speretta, A. Menicucci, and E. Gill. 2020. Utility and constraints of pocketqubes. *CEAS Space Journal*, 1–14.

[4] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze. 2017. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *Solid-State Circuits*.

[5] A. Colin and B. Lucia. 2016. Chain: tasks and channels for reliable intermittent programs. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 514–530.

[6] A. Colin, E. Ruppel, and B. Lucia. 2018. A reconfigurable energy storage architecture for energy-harvesting devices. In *Architectural Support for Programming Languages and Operating Systems.*

[7] B. Denby and B. Lucia. 2019. Orbital edge computing: Machine inference in space. *IEEE Computer Architecture Letters*.

[8] B. Denby and B. Lucia. 2020. Orbital edge computing: Nanosatellite constellations as a new class of computer system. In *Architectural Support for Programming Languages and Operating Systems*, 2020.

[9] H. Desai and B. Lucia. 2020. Power-aware heterogeneous architecture scaling for energy-harvesting computers. *IEEE Computer Architecture Letters*.

[10] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam. 2015. Shidiannao: Shifting vision processing closer to the sensor. In *ACM SIGARCH*.

[11] C. Foster, J. Mason, V. Vittaldev, L. Leung, V. Beukelaers, L. Stepan, and R. Zimmerman. Mar. 2018. Constellation Phasing with Differential Drag on Planet Labs Satellites. *Journal of Spacecraft and Rockets*, 55(2):473– 483.

[12] A. Gatherer and Z. Manchester. Aug. 2019. Magnetorquer-Only Attitude Control of Small Satellites Using Trajectory Optimization. In *AAS/AIAA Astrodynamics Specialist Conference*, Portland, ME.

[13] G. Gobieski, B. Lucia, and N. Beckmann. 2019. Intelligence beyond the edge: Inference on intermittent embedded systems. In *Proceedings of the International Symposium on Architecture Support for Programming Languages and Operating Systems.*

[14] G. Gobieski, A. Nagi, N. Serafin, M.M. Isgenc, N. Beckmann, and B. Lucia. 2019. Manic: A vector-dataflow architecture for ultra-low-power embedded systems. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 52*, 670684, ACM, New York, NY, USA.

[15] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M.A. Horowitz, and W.J. Dally. 2016. Eie: Efficient inference engine on compressed deep neural network. In *ISCA*. IEEE.

[16] S. S. E. Holdings. FCC Fixed Satellite Service Filing SAT-LOA-20161115-00118. 2016. Federal Communication Commission.

[17] S. S. E. Holdings. FCC Fixed Satellite Service Filing SAT-LOA-20170301-00027. 2017. Federal Communication Commission.

[18] S. S. E. Holdings. FCC Fixed Satellite Service Filing SAT-LOA-20170726-00110. 2017. Federal Communication Commission.

[19] A. Jain, Z. Kapetanovic, A. Kumar, V. N. Swamy, R. Patil, D. Vasisht, R. Sharma, M. Swaminathan, R. Chandra, A. Badam, et al., 2019. Low-cost aerial imaging for small holder farmers. In ACM *SIGCAS Conference on Computing and Sustainable Societies.*

[20] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R.G.L. D'Oliveira, S.E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascn, B. Ghazi, P.B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečny, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S.U. Stich, Z. Sun, A.T. Suresh, F. Tramr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F.X. Yu, H. Yu, and S. Zhao. 2019. Advances and open problems in federated learning.

[21] Katelyn Elizabeth Boushon. 2018. *Thermal analysis and control of small satellites in low Earth orbit.* (Master's thesis) Missouri University of Science and Technology.

THE FUTURE OF COMPUTATIONAL NANOSATELLITE CONSTELLATIONS

Satellites are becoming smaller, cheaper, simpler, and easier to launch en masse — creating an opportunity for computer systems researchers, application designers, and consumers. Emerging nanosatellite constellations make new applications possible by providing access to valuable orbital sensing data. Nanosatellites need computational capabilities and a high degree of autonomy. Designers of future nanosatellite missions are now forced to consider the constellation as a new type of distributed computer system. Treating a nanosatellite constellation as a distributed computer system presents new research challenges related to energy, constellation-scale distribution, secure edge multi-tenancy, geolocalized machine learning and inference, and the role of distributed motion planning and actuation in computer system design. The future is bright for research in computational space systems and orbital edge computing. As space becomes more accessible, the world is likely to see more innovative computer systems take flight. ■

Brandon Lucia is the Sathaye Family Career Development Associate Professor of Electrical and Computer Engineering at Carnegie Mellon University. His research interests lie at the intersection of programming languages, operating systems, and computer architecture, with a focus on ultra-low-power and intermittent computing, edge computer systems, and space-based computer systems.

Brad Denby is a PhD Candidate in the Electrical and Computer Engineering Department at Carnegie Mellon University. His research focuses on the intersection of computer systems and space systems with an emphasis on orbital edge computing.

Zachary Manchester is an assistant professor of Robotics at Carnegie Mellon University. He received a PhD in aerospace engineering from Cornell University and has led several small satellite missions. His research interest include numerical optimization, control, and estimation with applications to robotic and aerospace systems.

Harsh Desai is a PhD candidate in the Electrical and Computer Engineering Department at Carnegie Mellon University. His research targets the design and development of energy-harvesting sensor systems, with focus on improving end-to-end performance in variable energy-harvesting conditions.

Emily Ruppel is a PhD Candidate in the Electrical and Computer Engineering Department at Carnegie Mellon University. Her research touches hardware and software for low-power, energy-harvesting devices, with a focus on creating programming models for cyber-physical applications on intermittent systems.

Alexei Colin is a computer scientist at the University of Southern California Information Sciences Institute in Arlington, VA. His current work is in the domains of embedded systems for space applications and low-power circuits for energy harvesting and distribution.

[22] Kemet Electronics Corporation. 2020. Supercapacitors FT Series. https://content.kemet.com/datasheets/KEM_S6014_FT.pdf

[23] B. Klofas. Planet releases openlst, an open radio solution. 2018. https://www.planet.com/pulse/planet-openlst-radio-solution-for-cubesats/

[24] V. Kortbeek, K.S. Yldrm, A. Bakar, J. Sorber, J. Hester, and P. Paweczak. 2020. Time-sensitive intermittent computing meets legacy software. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS20, ACM, New York, NY, USA.

[25] B. Krashinsky, C. Batten, M. Hampton, S. Gerding, B. Pharris, J. Casper, and K. Asanovic. 2004. The vector-thread architecture. *IEEE Micro*, 24(6):84–90.

[26] D. Krejci and P. Lozano. Space propulsion technology for small spacecraft. 2018. *Proceedings of the IEEE.*

[27] L. Leung, V. Beukelaers, S. Chesi, H. Yoon, D. Walker, and J. Egbert. 2018. Adcs at scale: Calibrating and monitoring the dove constellation. In *Proc. AIAA/USU Conf. Small Satellites.*

[28] B. Lucia and B. Ransford. A simpler, safer programming and execution model for intermittent systems. 2015. In *ACM SIGPLAN Notices*, volume 50, 575–585. ACM.

[29] K. Maeng, A. Colin, and B. Lucia. 2017. Alpaca: Intermittent execution without checkpoints. In ACM *Proceedings of the 2017 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications.*

[30] K. Maeng and B. Lucia. 2020. Adaptive low-overhead scheduling for periodic and reactive intermittent execution. In ACM *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation.*

[31] Z. Manchester, M. Peck, and A. Filo. 2013. Kicksat: A crowd-funded mission to demonstrate the world's smallest spacecraft. In *Proceedings of the 27th AIAA/USU Conference, Small Satellite Constellations*, Logan, Utah, USA. SSC13-IX-5.

[32] F. Martel, P. Pal, and M. Psiaki. Active Magnetic Control System for Gravity Gradient Stabilized Spacecraft, 18.

[33] A. Mehrparvar, D. Pignatelli, J. Carnahan, R. Munakat, W. Lan, A. Toorian, A. Hutputanasin, and S. Lee. 2014. Cubesat design specification rev. 13. Technical report, California Polytechnic State University, San Luis Obispo.

[34] T. Nowatzki, V. Gangadhar, N. Ardalani, and K. Sankaralingam. June 2017. Stream-dataflow acceleration. *SIGARCH Comput. Archit. News*, 45(2):416429.

[35] NVIDIA. Nvidia jetson tx2/tx2i system-on-module data sheet. Technical report, NVIDIA, 2018.

[36] R. Prabhakar, Y. Zhang, D. Koeplinger, M. Feldman, T. Zhao, S. Hadjis, A. Pedram, C. Kozyrakis, and K. Olukotun. June 2017. Plasticine: A reconfigurable architecture for parallel patterns. *SIGARCH Comput. Archit. News*, 45(2):389402.

[37] S. Radu, M. Uludag, S. Speretta, J. Bouwmeester, A. Dunn, T. Walkinshaw, P. Kaled Da Cas, and C. Cappelletti. 2018. The pocketqube standard issue 1. Technical report, TU Delft.

[38] S. Resch, S.K. Khatamifard, Z.I. Chowdhury, M. Zabihi, Z. Zhao, H. Cilasun, J.P. Wang, S.S. Sapatnekar, and U.R. Karpuzcu. 2020. Mouse: Inference in non-volatile memory for energy harvesting applications. In 2020 *53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 400–414.

[39] E. Ruppel and B. Lucia. 2019. Transactional concurrency control for intermittent, energy harvesting, computing systems. In ACM *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation.*

[40] O. W. Satellites. 2016. FCC Fixed Satellite Service Filing SAT-LOI-20160428-00041. Federal Communication Commission.

[41] O. W. Satellites. 2018. FCC Fixed Satellite Service Filing SAT-MOD-20180319-00022. Federal Communication Commission.

[42] M. Surbatovich, B. Lucia, and L. Jia. Nov. 2020. Towards a formal foundation of intermittent computing. *Proc. ACM Program.* Lang., 4(OOPSLA).

[43] Tactical Technology Office. 2018. Broad Agency Announcement: Blackjack Pit Boss, HR001119S0012. Technical report, DARPA.

[44] TI Inc. MSP430FR599x, MSP430FR596x Mixed-Signal Microcontrollers. 2018. https://www.ti.com/lit/ds/symlink/msp430fr5994.pdf

[45] J. Van Der Woude and M. Hicks. 2016. Intermittent computation without hardware support or programmer intervention. In *Proceedings of OSDI'16: 12th USENIX Symposium on Operating Systems Design and Implementation*, 17.

[46] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman. 2017. Farmbeats: An IoT platform for data-driven agriculture. In *Symposium on Networked Systems Design and Implementation.*

[47] Zac Manchester. 2015. KickSat. http://zacinaction.github.io/kicksat/.