SSC13-IX-5

KickSat: A Crowd-Funded Mission To Demonstrate The World's Smallest Spacecraft

Zachary Manchester, Mason Peck
Cornell University
Upson Hall, Ithaca, NY 14853; 607-279-1358
zrm3@cornell.edu

Andrew Filo
4Special Projects
22670 Oakcrest Ct, Cupertino, CA 95014; 650-940-1677
afilo@earthlink.net

ABSTRACT

Thanks to rapid advances made in the semiconductor industry, it is now possible to integrate most of the features of a traditional spacecraft onto a chip-scale device. The Sprite ChipSat, in development at Cornell since 2008, is an example of such a device. The KickSat mission, scheduled for launch in late 2013, will deploy 128 Sprites in low Earth orbit to test their survivability and demonstrate their code division multiple access (CDMA) communication system. The Sprites are expected to remain in orbit for several days while downlinking telemetry to ground stations before reentry. KickSat has been partially funded by over 300 backers through the crowd-funding website Kickstarter. Reference designs for the Sprites, along with a low-cost ground station receiver, are being made available under an open-source license.

INTRODUCTION

The rapid miniaturization of commercial-off-the-shelf (COTS) electronics, driven in recent years by the emergence of smart phones, has made many of the components used in spacecraft available in very small, low-cost, low-power packages. This has inspired the "ChipSat" concept[1] – the idea of building a chip-scale satellite using the same devices and processes used in the consumer electronics industry. The ability to mass produce such devices, along with their small size, leads to the realistic near-term possibility of sub-$1,000 dollar-per-satellite missions for scientific, educational, and hobbyist use.

A new class of science missions will be possible thanks to ChipSat "sensor clouds." Large ensembles of ChipSats equipped with a range of different electromagnetic, micro-electro-mechanical (MEMS), and nanofluidic sensors could enable, for example, large-scale in-situ surveys of planetary atmospheres[2] or asteroid surface composition[3]. Earth's ionosphere, in particular, could be studied in ways that are impractical or impossible with current sounding rocket and multi-satellite missions[4,5]. ChipSat sensor clouds would uniquely allow thousands of data points to be collected simultaneously over large spatial volumes while also providing a high degree of robustness to individual spacecraft failures.

By dramatically reducing the cost and complexity of building and launching a spacecraft, ChipSats could also help expand access to space for students and hobbyists. In the near future, it will be possible for a high school science class, amateur radio club, or motivated hobbyist to choose sensors, assemble a ChipSat, configure a small ground station, and fly their own satellite mission.

The KickSat project was founded at Cornell in 2011 with the goal of advancing the core technologies needed to enable low-cost ChipSat missions. Over the past two years, the Sprite ChipSat, a CubeSat-based deployer for the Sprite, and a software-defined radio (SDR) ground station have been developed. The hardware designs, code, and documentation for all of these systems are being released under open-source licenses where possible[6]. KickSat's first orbital demonstration, in which 128 Sprites will be deployed in low Earth orbit, is scheduled for launch in late 2013 as a secondary payload on the SpaceX CRS-3 mission.

Much of the funding that has made KickSat possible was raised through the crowd-funding website Kickstarter[7] (Figure 1). Over 300 backers pledged individual amounts from $25 to $10,000 in exchange for rewards ranging from having their names silkscreened onto articles of flight hardware to being able to trigger the deployment of the Sprites along with

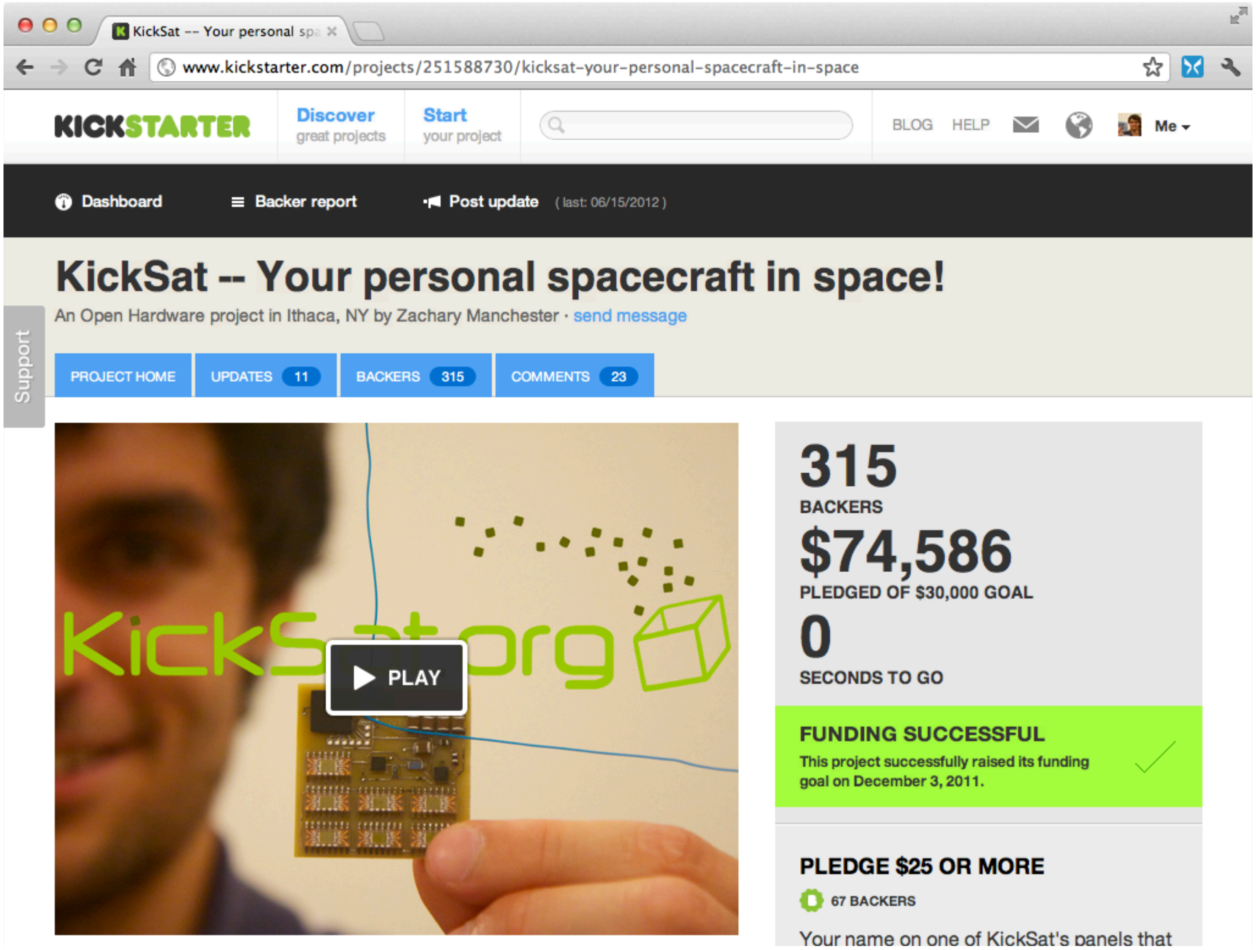

Figure 1: Kickstarter Fundraising Website

the ground control team. In a particularly interesting foreshadowing of the possibilities for hobbyist space missions, 26 individuals pledged $1,000 in exchange for a development kit to write flight code for their own ChipSat. In total, nearly $75,000 was raised between October and December 2011, as shown in Figure 2.

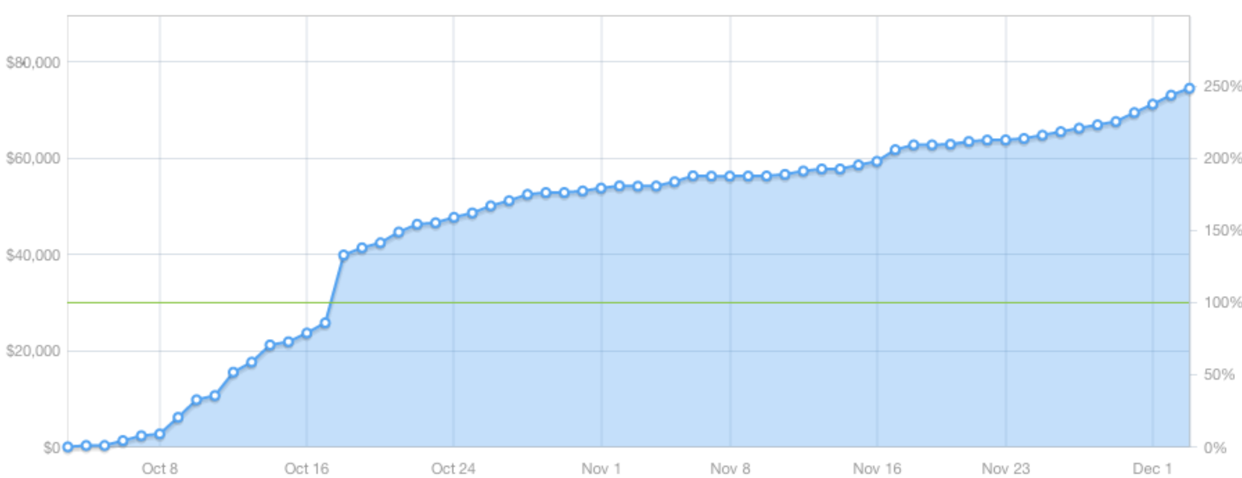

Figure 2: Fundraising Progress

The remainder of this paper describes the Sprite ChipSat and gives an in-depth overview of its code division multiple access (CDMA) communication system, followed by a discussion of the deployment system and an overview of the planned KickSat mission.

THE SPRITE CHIPSAT

The Sprite makes use of modern, low-cost, low-power integrated circuits to create a general purpose "spacecraft bus" for chip-scale sensors. It includes Spectrolab TASC solar cells, a Texas Instruments CC430 microcontroller and radio system-on-chip (SoC), a Honeywell HMC5883L 3-axis magnetometer, and an InvenSense ITG-3200 3-axis MEMS gyro, as well as associated passive components, on a printed circuit board measuring 3.5 by 3.5 centimeters with a mass of 5 grams (Figure 3).

The CC430 SoC is the core of the Sprite, providing all computing and communication capabilities. It combines an MSP430 microcontroller, which is clocked at 8 Mhz and provides 4 kB of RAM and 32 kB of flash memory, with a very flexible CC1101 UHF transceiver capable of output powers up to 10 mW and data rates up to 500 kbps. Both the MSP430 and CC1101 have flight heritage on CubeSat missions. An Arduino-based development environment, known as Energia, has been ported to the CC430 to facilitate rapid code development and prototyping[8].

The Sprite's antenna is a half-wave V-dipole, chosen for its isotropic gain pattern, easy tuning, and 50 ohm characteristic impedance, which eliminates the need for a matching network or balun circuit. The antenna is made of nitinol, a nickel-titanium alloy commonly referred to as "shape-memory alloy" or "memory metal," which can be deformed to a tremendous degree and still return to its original shape. Nitinol was chosen so that the antenna could be coiled within the small footprint of the Sprite PCB and still return to its intended geometry upon deployment.

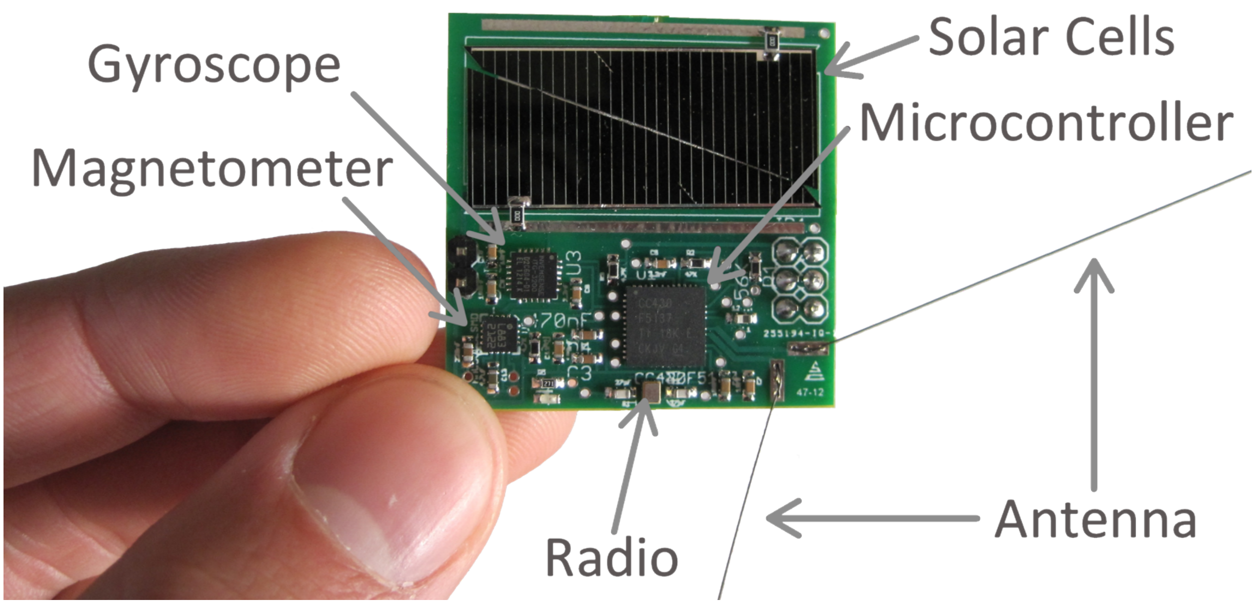

Figure 3: Sprite ChipSat

The components on the Sprite were chosen with the goal of minimizing size and maximizing functionality within the available power budget. The solar cells deliver up to 60 mA of current at 2.2 volts directly to the electronics with no energy storage or power conditioning. The maximum current consumption of the Sprite is approximately 35 mA, so there is considerable margin. Several lithium-ion and lithium-polymer batteries were investigated that fit the size, mass, and power constraints of the Sprite, but it was determined that none could survive the low temperatures encountered during eclipse. Therefore, the Sprites can operate only in sunlight and will be completely un-powered over the night-side of the Earth.

CLOSING THE LINK

Perhaps the most difficult engineering challenge associated with the KickSat project is closing the communication link from orbiting Sprites to ground stations. The Sprite's transmitter is limited to about 10 mW of output power. A lack of closed-loop attitude control means a low-gain antenna with an omnidirectional gain pattern is required. Additionally, due to licensing constraints, the Sprites must efficiently share limited RF bandwidth. Closing link over several hundred kilometers with all of these constraints is a formidable challenge, but thanks again to advances in consumer electronics, it turns out to be possible with relatively inexpensive hardware.

As motivation, we first consider the signal-to-noise ratio (SNR) for a Sprite signal as seen by a ground station. With a transmitter power $P_t = 10\,\text{mW} = 10\,\text{dBm}$, a transmitter antenna gain of $G_t = 0\,\text{dB}$, receiver antenna gain $G_r = 7\,\text{dB}$, distance R = 500 km, and wavelength $\lambda = 70\,\text{cm}$, the received power $P_r \approx -122\,\text{dBm}$ is given by the Friis equation[9]:

$$P_r = P_t + G_t + G_r + 20 \cdot \text{Log}_{10}\left(\frac{\lambda}{4\pi R}\right) \quad (1)$$

The noise power in the receiver $P_n \approx -120\,\text{dBm}$ is given by equation 2[9],

$$P_n = F_N + 10 \cdot \text{Log}_{10}(K_B T B) \quad (2)$$

where $F_N = 9\,\text{dB}$ is the receiver's noise figure, K_B is Boltzmann's constant, $T = 150\,\text{K}$ is the background temperature in Kelvin, and $B = 64\,\text{kHz}$ is the receiver bandwidth. Subtracting the noise power from the signal power gives an SNR $= P_r - P_n$ of -2 dB.

It is worth pausing here to consider the implications of the previous result: A negative SNR means, in a power sense, there is more noise than signal. If one were to connect an oscilloscope between the antenna and receiver, the observed signal would be nearly indistinguishable from white noise. In a more typical situation, a design engineer might use a more powerful transmitter or higher gain antennas, but in the case of the Sprite, neither of those are options.

Communication Background

The Sprite communication system makes use of two techniques, matched filtering and forward error correction (FEC), which have long histories in space and terrestrial communication, navigation, and radar systems but, unfortunately, have not yet seen wide use within the small satellite community. This section will provide a brief overview of these techniques for the aerospace engineer with limited signal processing experience. It is the authors' view that more efficient utilization of communication bandwidth with techniques like these would enable far greater data throughput from small satellite missions.

To overcome low SNR, matched filtering, which is the optimal linear filter for maximizing SNR[10], is used. The basic idea is to substitute each data bit with a long, specially chosen string of bits known as a pseudo-random number (PRN) code that is agreed upon by the transmitter and receiver *a priori*. In this context, the bits making up the PRN code are commonly referred to as "chips" to differentiate them from the data bits. Rather than attempting to lock onto the carrier and demodulate the chips individually, the receiver instead looks for the entire PRN code by cross-correlating the incoming signal against the known code at each time step.

Cross-correlation in the discrete-time setting can be interpreted as a "sliding inner product," as shown in equation 3, where x is the cross-correlation, p is the sampled PRN code vector of length N, and s is a vector of signal samples. At each time step k, the signal vector is shifted one sample, with the oldest element being removed from one end and a new sample being added at the opposite end, then a new inner product is calculated. If the PRN is present, the correlation will be high, even in the presence of substantial noise, while if no code is present, the correlation will be low.

$$x_k = p^\dagger \begin{bmatrix} s_{k-N} \\ \vdots \\ s_k \end{bmatrix} \quad (3)$$

This simple form of the cross-correlation is essentially the one used in the Sprite receiver, with an additional frequency search step to detect and correct for Doppler shift. Cross-correlation can also be calculated in terms of the Fourier transform, and with a good Fast Fourier Transform implementation, that method is often computationally faster[11].

Matched filtering essentially allows the energy in the entire PRN code to be summed and treated as a single data bit, providing an increase in SNR equal to the code length. The trade off, however, is that the data rate is also lower by the same factor. For the KickSat mission, a family of PRN codes 640 chips long is being used, providing a "code gain" $G_c = 10 \cdot \text{Log}_{10}(640)$ of about 28 dB for a very robust link margin.

Aside from improving SNR, matched filtering also makes possible code division multiple access (CDMA). By assigning each Sprite a different PRN, the receiver can "tune" to a particular spacecraft's signal by correlating against its unique code. This allows all the Sprites on a particular mission to share the same allocated frequency, simplifying licensing and eliminating the need for clock synchronization that would otherwise be required for the Sprites to alternate transmitting on the same frequency. CDMA has been used for many years in the cellular telephone industry[10] and the global positioning system[12], and has proven in practice to be the most efficient channel access method when a large number of users must be accommodated[10].

To successfully implement CDMA, the family of PRN codes assigned to the group of Sprites must be carefully chosen to minimize interference. They must be as nearly orthogonal as possible in the sense that their cross-correlations should be close to zero. Unfortunately, it is mathematically impossible to generate code families with perfect zero cross-correlations for all time offsets[13]. Many code families exist, however, with low and bounded cross-correlations, notably the Gold Codes[12] used in GPS. For the KickSat mission, a code family based on SLCE sequences[14] has been selected which allows greater flexibility in choosing code lengths.

Due to the large number of Sprites being deployed by KickSat and the finite cross-correlations between PRN codes, all of the Sprites cannot transmit simultaneously without causing unacceptable levels of interference. To overcome this, the Sprites operate their radios on a 5% duty cycle with randomized sleep and wake times. Because excessive interference may still occur occasionally, forward error correction is applied to further improve robustness. FEC is widely used in modern digital communication because it allows a receiver to correct errors in a message without having to request a re-transmission. The idea is to pad the message with extra bits, known as parity bits, based on a mathematical rule. For the Sprite, a linear block code is used where the parity bits are generated by simple matrix multiplication.

To encode an 8-bit message byte m, the Sprite encoder treats the byte as an 8-dimentional binary vector and multiplies it by a 16-by-8-bit matrix G, known as the generator matrix of the code, producing a 16-bit code word c (Equation 4).

$$c = m \cdot G \tag{4}$$

Note that the convention in coding theory is to use row vectors[15]. Multiplication in this context, where vectors are over the field of binary numbers (known as a Galois Field and abbreviated GF(2) in coding theory), is equivalent to the exclusive-or (XOR) operation from Boolean logic.

In the notation commonly used in coding theory, this is a (16,8,5) block code, where 16 is the code word length, 8 is the message length, and 5 is the Hamming distance[15], which determines the ability of the code to correct errors. With a Hamming distance of 5, up to 2 bit flips or 5 bit erasures can be corrected[15].

There are several ways a receiver can decode a block code like the one presented here. In the Sprite receiver, a simple brute-force soft decoder is used, where the received code word is compared to all 256 possible code words. An inner product is calculated with each one, and the decoded byte is taken to be the best match. While this decoder is optimal in the sense that it produces the maximum-likelihood message byte, its computational complexity scales exponentially with the message length and it quickly becomes intractable for larger codes. Several algebraic decoding methods exist[10,14] which are, in general, sub-optimal, but which are much more computationally efficient.

Ground Station Receiver

With the goal of enabling as many people as possible to participate in the KickSat project, a reference design for a low-cost and portable Sprite receiving station has been developed. The hardware consists of a hand-held Yagi antenna, a low-noise amplifier (LNA), a low-cost USB radio receiver dongle (commonly known as a DVB-T or RTL dongle), and a PC running the GNU Radio software[16] (Figure 4). The hardware is widely available from amateur radio suppliers and online retailers at a total cost, not including the PC, of about $200. Full instructions for assembling a ground station will be made available online on the KickSat project wiki[6].

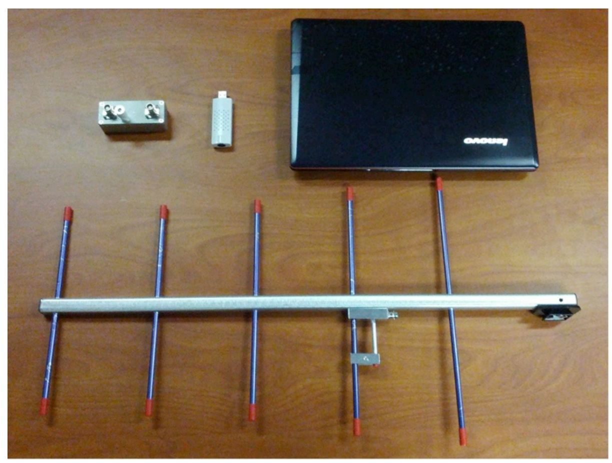

Figure 4: Ground Station Hardware

Because of the signal processing requirements inherent in the Sprite receiver design, a software defined radio (SDR) architecture was chosen. The DVB-T dongle functions as a tuner and analog to digital converter, bringing the raw baseband signal into the PC. From there, the rest of the receiver is written in C++ as a set of blocks for the GNU Radio software framework. Figure 5 shows a screenshot of the receiver software with the signal flow block diagram at the top of the window.

Starting from the DVB-T dongle input on the left (the block labeled “RTLSDR Source” in the diagram), the

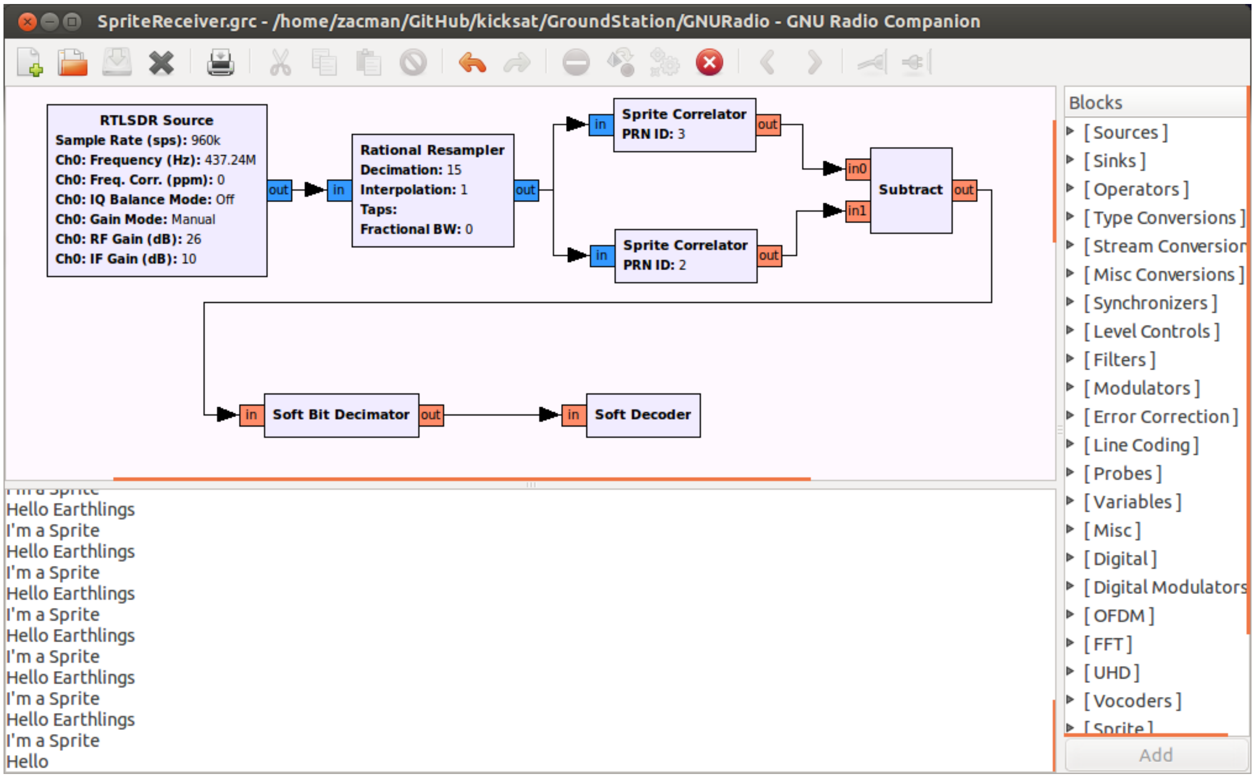

Figure 5: GNU Radio Receiver

signal is decimated (low-pass filtered and down-sampled) to one sample per PRN chip, which is 64 kHz in the current implementation. From there, it passes through two PRN correlators, each of which performs matched filtering against a different code. Figure 6 shows the output of a correlator in which the spike corresponding to a detected PRN code is clearly visible.

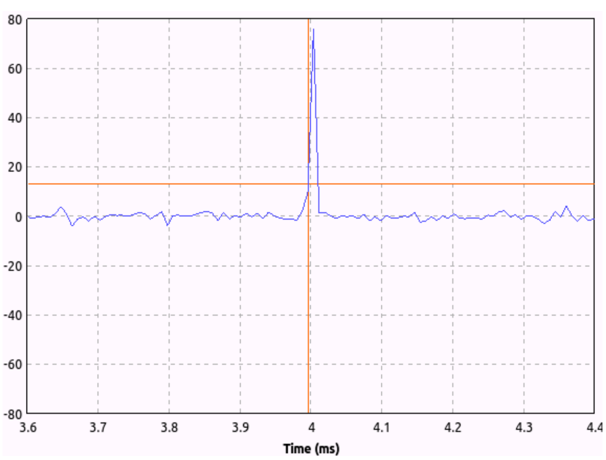

Figure 6: Correlator Output

Each Sprite is assigned two PRNs – one corresponding to a zero bit and the other corresponding to a one bit. The correlator outputs are subtracted, resulting in a zero-mean signal where a positive spike corresponds to a one and a negative spike corresponds to a zero. The signal is then down-sampled again, this time to the bit rate (100 Hz), by the Soft Bit Decimator block, before passing into the decoder. The decoder multiplies the received code word vector by a matrix whose columns consist of all possible valid code words. The resulting vector is searched for its largest value, which corresponds to the most likely message byte. Finally, the resulting byte is written to the console, as seen at the bottom of Figure 5.

The Sprite software receiver can run in real time on relatively recent PC hardware. It can also run in a batch mode where RF data is recorded during a pass and fed through the receiver later, when the signals of each Sprite can be extracted individually. Both scenarios have been successfully tested outdoors with Sprites and receiver separated by 25 miles and an additional 23 dB of attenuation inserted after the receiver antenna, roughly corresponding to the link conditions between LEO and Earth stations anticipated for the KickSat mission.

THE KICKSAT SPACECRAFT

KickSat is a 3U CubeSat consisting of a 1U bus and a 2U Sprite deployer (Figure 7). The bus is based on the flight-proven PhoneSat[17] 2.0 CubeSat developed at NASA Ames Research Center and is being built using primarily COTS components to provide power, communication, command and data handling, and attitude determination and control functions.

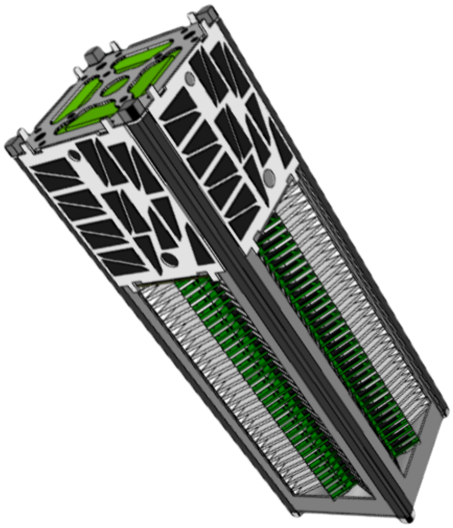

Figure 7: KickSat Spacecraft

Sprite Deployer

The Sprite deployer has been designed for simplicity and robustness while minimizing attitude disturbances on the spacecraft during deployment. The deployer contains 128 Sprites stacked in four columns in a 2-by-2 arrangement. Each Sprite is housed in an individual slot and constrained by a carbon fiber rod that runs the length of the column, passing through a hole in the corner of every Sprite (Figure 8). The nitinol wire antennas on the Sprites are coiled in such a way that they act as springs, pushing the Sprites out of their slots.

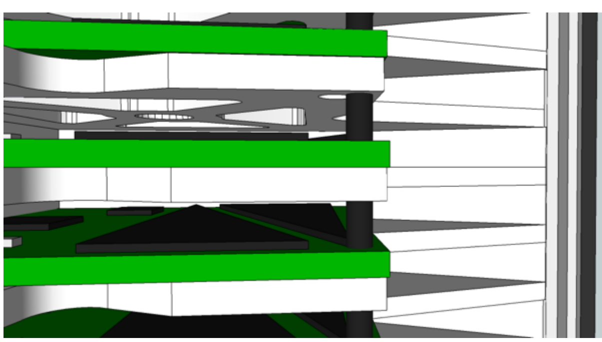

Figure 8: Sprites in Deployer

All four carbon fiber rods are attached to a single plate at the end of the deployer that is actuated by a compressed spring and held in place by a locking mechanism, shown in Figure 9.

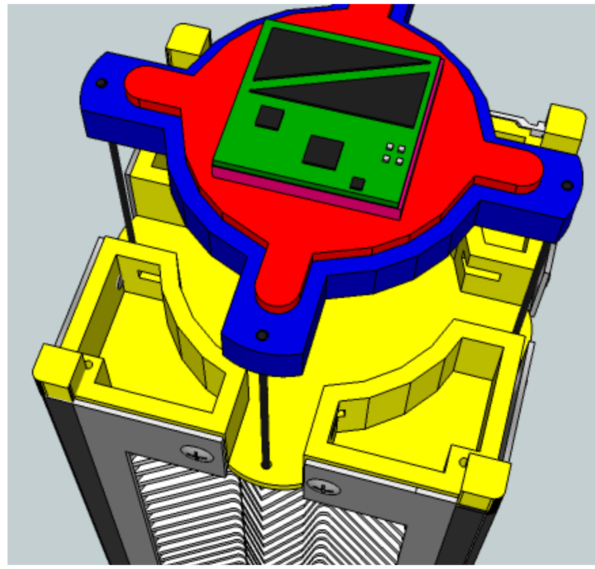

Figure 9: Deployer Locking Mechanism

Deployment is triggered by a nichrome burn wire, which unlocks the mechanism, allowing the spring to pull the four rods out. The Sprites' antennas then push them from the deployer housing, as shown in Figure 10, with an estimated ΔV of 5-10 cm/sec.

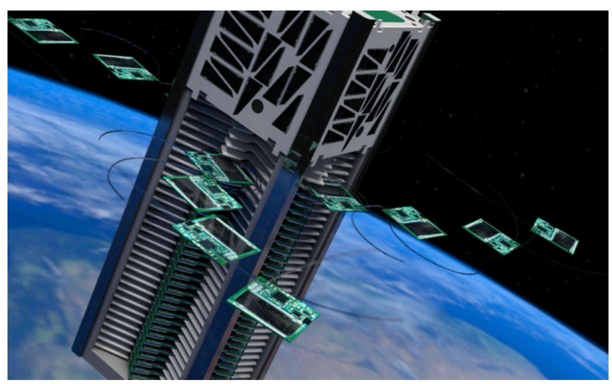

Figure 10: Deployment
Image courtesy of Ben Bishop

MISSION PROFILE

KickSat's launch has been awarded through NASA's Educational Launch of Nanosatellites (ELaNa) program, which places university-built CubeSats as secondary payloads on NASA missions. KickSat is currently manifested on CRS-3, a Space-X Falcon 9 set to launch in late 2013. CRS-3's primary mission is to bring supplies to the International Space Station. So, KickSat will be placed in roughly the same orbit as the ISS: a 325 km altitude circular orbit with an inclination of 51.5°. This orbit has two useful properties. First, the high inclination will put the Sprites in view of almost every populated area of the Earth during the mission, allowing many people to receive Sprite signals. Second, the low altitude limits the orbital lifetime of the Sprites to a few days, mitigating orbital debris concerns.

Upon separation from the launch vehicle, KickSat will power up and start a countdown timer. At 30 minutes, the bus UHF radio antenna will be deployed, then at 45 minutes the bus radio will begin transmitting a beacon signal. During the first few passes, ground station operators will establish communication and perform checkouts of the spacecraft. Over the next three to four days, the attitude control system in the bus will be used to point KickSat's minor axis of inertia (long axis) at the sun, and then spin the spacecraft up to 10-15 RPM about the sun vector, ensuring attitude stability during the deployment sequence.

Once a stable sun-pointing attitude has been established, all systems have been checked out, and KickSat is in view of a ground station, a signal from the ground will trigger a nichrome burn wire to unlock the deployer. The deployer's spring mechanism will then release the Sprites as free-flying spacecraft. The deployment sequence is intended to release the Sprites in a sun-pointing major-axis spin, which is dynamically stable[17]. While there are no strict pointing requirements, the goal of this spin stabilization is to keep the Sprites' solar panels pointed largely at the sun for the short mission lifetime and to minimize nutation, which would otherwise introduce attitude-dependent fluctuations in power.

Upon being exposed to sunlight, the Sprites will immediately be powered on and will begin taking sensor measurements and transmitting telemetry. Cornell will operate the primary ground station in Ithaca, New York, however amateur radio operators are being encouraged to set up their ground stations based on the published reference design. The primary criterion for mission success is the reception of data packets from the Sprites.

Due to their extremely low ballistic coefficient, the Sprites are expected to remain in orbit for only a few days before reentering and burning up in the atmosphere, alleviating debris concerns. Figure 11 shows altitude vs. time plots for bounding maximum and minimum, as well as average drag cases using the MSIS atmospheric model[9].

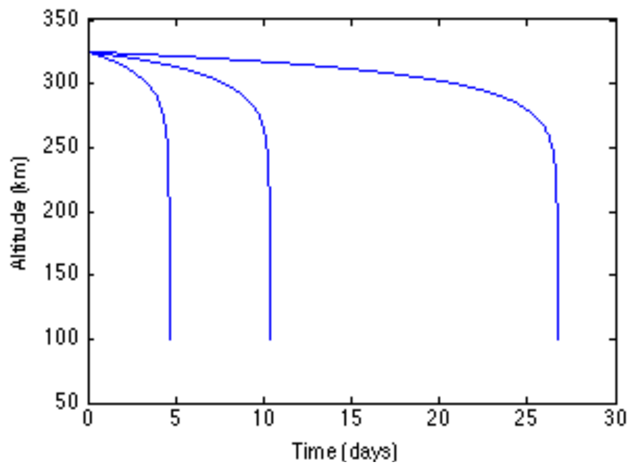

Figure 11: Altitude vs. Time for a Sprite in LEO

In line with other CubeSat missions, the KickSat bus is expected to remain in orbit for a few months before it too reenters. It may serve as a test bed for further communication and attitude control experiments during that time.

CONCLUSION

The KickSat project seeks to make access to space faster, cheaper, and more accessible. To ensure that the technologies developed for KickSat are available to as broad an audience as possible and to enable further research, KickSat is an open source project. All of the design files, code, and documentation being developed will be made available online, free of charge, for anyone to build or use as a starting point for new designs.

KickSat represents the beginning of a new paradigm for low-cost space missions. The small size, mass production, redundancy, and short development times possible with ChipSats will enable new science missions, as well as expand space access to greater numbers of people. In the near future, it will be possible for scientists, students, and hobbyists alike to put together ChipSat-based missions in a matter of weeks or days instead of the months or years common with current small satellite missions today, and at one to two orders of magnitude less cost.

ACKNOWLEDGEMENTS

The authors would like to thank the many people who have financially supported KickSat through Kickstarter, as well as the members of the small satellite community at NASA Ames Research Center for their continued enthusiasm and support.

REFERENCES

1. Atchison, J.A. and M.A. Peck, "A Millimeter-Scale Lorentz-Propelled Spacecraft," AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, SC, August 20-23, 2007.
2. Atchison, J.A., Z.R. Manchester, and M.A. Peck, "Microscale Atmospheric Reentry Sensors," 7th International Planetary Probe Workshop, Barcelona, Spain, June 14-18, 2010.
3. Manchester, Z.R. and M.A. Peck, "Stochastic Space Exploration with Microscale Spacecraft," AIAA Guidance, Navigation, and Control Conference, Portland, OR, August 8-11, 2011.
4. Escoubet, C., R. Schmidt, and M. Goldstein, "Cluster – Science and Mission Overview," Space Science Reviews Vol. 79, No. 1-2, 1997.
5. Angelopoulos, V., "The THEMIS Mission," Space Science Reviews Vol. 134, No. 1-4, 2008.
6. Manchester, Z.R., "KickSat Wiki," URL: http://github.com/zacinaction/kicksat/wiki [cited 8 June 2013].
7. Manchester, Z.R. and M. Johnson, "KickSat – Your personal spacecraft in space!" URL: http://www.kickstarter.com/projects/251588730/kicksat-your-personal-spacecraft-in-space [cited 10 June 2013].
8. Wessels, R., et. al. "Energia," URL: www.energia.nu [cited 8 June 2013].
9. Wertz, J.R. and W.J. Larson, "Space Mission Analysis and Design," 3rd ed., Microcosm Press, 1999.
10. Sklar, B., "Digital Communications," 2nd ed., Prentice Hall, 2000.
11. Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, "Numerical Recipes," 3rd ed., Cambridge University Press, 2007.
12. Misra, P. and P. Enge. "Global Positioning System: Signals, Measurements, and Performance," Revised ed., Ganga-Jamuna Press, 2010.
13. Welch, L.R., "Lower Bounds on the Maximum Cross Correlation of Signals," IEEE Trans. on Info. Theory, vol. 20, no. 3, pp. 397–399, May 1974.
14. Wallner, S. and J. Avila-Rodriguez,, "Codes: The PRN Family Grows Again," Inside GNSS, Septeber 20ll.
15. Moon, T.K., "Error Correction Coding," 1st ed., Wiley, 2005.
16. Blossom, E., "GNU Radio: Tools for Exploring the Radio Frequency Spectrum," Linux Journal, URL: http://www.linuxjournal.com/article/7319 [cited 11 February 2013].
17. "PhoneSat: NASA's Smartphone Nanosatellite," URL: www.phonesat.org [cited 8 June 2013].
18. Hughes, P.C., "Spacecraft Attitude Dynamics," Revised ed., Dover, 2004.