Rapid Development of a Balloon-Borne CDH System using COTS Electronics and Open Source Software

Zach Dischner
Southwest Research Institute
1050 Walnut St.
Boulder, CO 80305
303-546-9670
dischnerz@boulder.swri.edu

Kevin Dinkel
Southwest Research Institute
1050 Walnut St.
Boulder, CO 80305
303-546-9670
dinkelk@boulder.swri.edu

Jed Diller
Southwest Research Institute
1050 Walnut St.
Boulder, CO 80305
303-546-9670
dillerj@boulder.swri.edu

Nick Truesdale
Southwest Research Institute
1050 Walnut St.
Boulder, CO 80305
303-546-9670
truesdalen@boulder.swri.edu

Abstract— **Open source tools and readily available consumer hardware offer many advantages over the typical, custom designs used for control and data handling (CDH) and electronic power systems (EPS), especially in applications where the development time is measured in months, not years. This paper highlights the development of the CDH and EPS systems for the UV-Visible optics bench onboard the Balloon Rapid Response to Comet ISON (BRRISON) balloon gondola, a 5 month effort from conception to flight. The system was designed to power, operate, and collect data from an optical bench which included a fine-steering mirror motion compensation system, one high speed scientific CMOS camera, a science-grade CCD, a fold mirror mechanism, a filter wheel, and an active thermal heating element. To enable rapid development of a system with such complex and diverse requirements, a strategy was employed to use open source and consumer products. SwRI constructed a flight-ready power system designed around an Arduino microcontroller board, and a data collection system which includes a desktop motherboard running Ubuntu Linux onboard a recent Intel processor. The result is a highly portable, modular system which is efficient and inexpensive to reproduce for future high altitude balloon missions. This paper discusses the benefits and challenges SwRI faced using COTS (commercial off-the-shelf) hardware and open source software for rapid development.**

TABLE OF CONTENTS

1. INTRODUCTION 1
3. CDH SUBSYSTEM 2
2. EPS SUBSYSTEM..6
4. CONCLUSION 8
REFERENCES...9
BIOGRAPHY...9

1. INTRODUCTION

BRRISON (Balloon Rapid Response to ISON) is a NASA-funded high altitude balloon mission, whose science goals are to obtain infrared, ultra-violet, and visible imagery of the comet ISON, as well as characterize the performance of a fine-steering mirror motion compensation system. Figure 1 shows the balloon gondola, telescope, and two optical benches, housing the mission's scientific instruments. The BIRC (BRRISON Infrared Camera) bench houses an infrared detector that was designed and built by Johns Hopkins APL. The UVVIS (UV-Visible) bench, which supports the ultra-violet and visible imaging systems as well as the FSM (fine-steering-mirror), was designed and fabricated by the SwRI (Southwest Research Institute) in Boulder, Colorado.

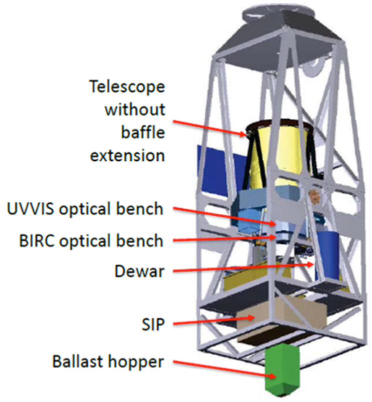


Figure 1: The BRRISON balloon gondola was a joint effort between APL and SwRI.[1]

Comet C/2012 S1 (ISON) was discovered on September 21, 2012. A study for the feasibility of the BRRISON project was launched shortly after, in December 2012. The project was accepted and awarded in February 2013 for a projected launch date in September of 2013. To meet the launch date, it was determined that the instrument needed to be "hands-off" at the beginning of August, leaving just 5 months to develop the entire UVVIS optical bench, supporting electronics, and software.[2] The focus of this paper is on the design and implementation of the power electronics (EPS) and the control and data handling (CDH) system for the UVVIS bench.

The Southwest Research Institute Planetary Science Division in Boulder, CO has minimal previous experience working on balloon flight electronics and software systems of this magnitude. With no legacy personnel, no legacy hardware, and no legacy software designs, SwRI had to design the electronics and software components from the ground-up. However, this clean-slate provided flexibility in design not usually experienced on comparable missions.

978-1-4799-1622-1/14/$31.00 ©2014 IEEE

This paper outlines the EPS and CDH systems with a focus on how each met specification, despite aggressive time constraints, by employing the use of open source tools and COTS (commercial off-the-shelf) hardware. A summary of the development experience is provided, highlighting what worked well and lessons learned.

3. CDH SUBSYSTEM

The UVVIS CDH subsystem is responsible for operating and controlling the UVVIS bench, as well as communicating with the gondola main computers. The design of the CDH subsystem began around a collection of optical bench components chosen by mechanical and optical engineers on the project.

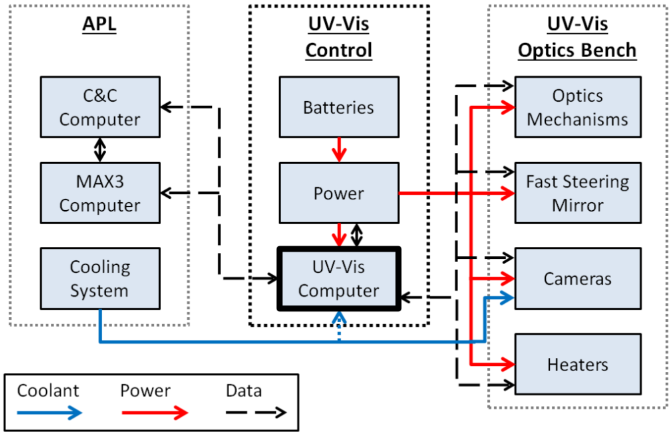


Figure 2: The CDH block diagram above shows the variety of interfaces the system had to control.

Figure 2 shows the components that the CDH system would need to integrate together. The primary design driver was to take images from two onboard cameras and use them in conjunction with a fine-steering mirror controller to stabilize the field of view. In addition, interface requirements included controlling and monitoring two motorized mechanisms, running a heater controller, and communicating with the gondola network.

The primary science camera was a Princeton Instruments PRO-EM 1024B, which acquires 1-megapixel UV and visible science images. The guide camera was an Andor Zyla, capable of obtaining 5-megapixel images at 100fps, sustained indefinitely. It acquires visible spectrum science images, and its frame rate is fast enough to be effective in a closed-loop control system with a fine-steering-mirror.

The most complex component, the fine steering mirror image stabilization system, drove the need for computational performance. The system had to be capable of obtaining 5-megapixel images from the guide camera at a speed of at least 50Hz and analyze these images in real time. Error signals from the analysis would then need to be relayed to the FSM controller with minimal latency in order to correct disturbances. In addition to acting on the images in real time, there was a requirement to save all acquired images in a lossless format. This proves a non-trivial task when a single camera is capable of producing and transferring over 1 GB of raw data per second. Interfaces to the rest of the peripheral components were predefined in the form of Ethernet, USB, serial, and specialized PCIe cards. As a result of speed, storage, and interface requirements, the standard design of miniature computer hardware typically implemented on balloon platforms would not suffice. The UVVIS electronics box would need to perform at the level of high-end consumer desktops, but be made to perform at the edge of space.

All this functionality needed to be contained with a volume of 16.5” x 14” x 8” and weigh no more than 20 kg. The entire system, electronics, cameras, mechanisms, and heaters included, had to run on less than 150 watts of average power and perform in any physical orientation, attached to the back of a moving telescope.

With the requirements laid out, 5 months till launch, and only 4 part-time workers tasked on electronics and software, it was decided that the best design approach was to have a single central computing architecture. By using a single multi-purpose computer instead of many smaller scoped processors, SwRI was able to maintain a single code base, centralize the integration effort, and avoid adding more interfaces to an already heavily IO-bound problem. The challenge in using a single system to do so many different things concurrently was that the system needed to have an extremely high level of performance. To keep things simple, cost effective, and timely, a high end consumer desktop retrofitted for balloon flight was the approach SwRI took.

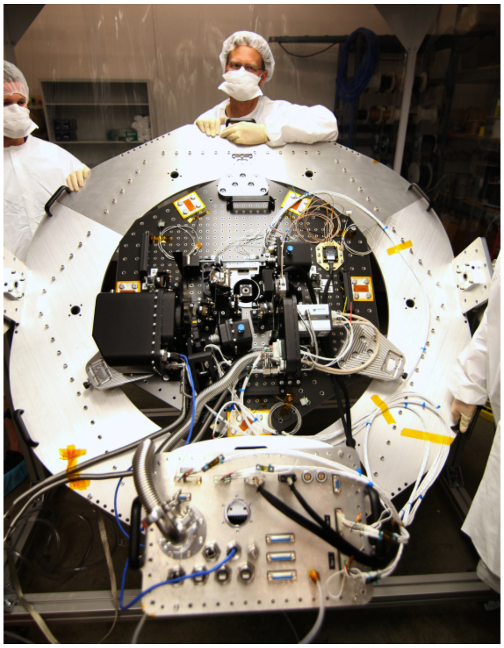


Figure 3: The UVVIS bench, shown above, includes two cameras, a fold mirror mechanism, a filter wheel, and a multitude of sensors. All of these devices are electrically

connected to the pressure vessel mounted on the bottom.

COTS Hardware Challenges

Choosing COTS hardware in the form of a high-end consumer desktop computer for engineering work has the advantages of being flexible in terms of processing power, modularity, and accessibility. Installing and testing a variety of processors, networking interfaces, storage devices, and general IO is a trivial matter. Nearly all components of the CDH flight computer can be found from electronic distribution giants such as Newegg. This proved invaluable throughout the testing and integration process. When replacement parts or additional components were required, they could be obtained overnight and installed and tested the next afternoon. Figure 4 shows the overnighted delivery of the entire first-revision CDH system, and some necessary peripherals.

Figure 4: The first order of the CDH COTS components from Newegg is displayed here in the SwRI optics lab.

Table 1: Computer Build List

Component	Hardware
Motherboard	ASUS P8 Z77 WS
CPU	Intel i7 3770K
Ram	32Gb Corsair Dominator Platinum 2400GHz
Storage	6x OCZ Vertex 512Gb SSD
Networking	4x Gigabit Ethernet
Analog, Digitial IO	Sensoray Model 826 PCIe card
Serial Communication	2x USB2 to RS232 Adapter 1x USB2 to RS485 Adapter
Fan	Noctua NH-L12 Heatsink

The first challenge SwRI faced was how to physically mount all the hardware within a volume of roughly a cubic foot. High Precision Devices (HPD) was contracted with manufacturing the pressure vessel to house the flight electronics. Maximum pressure vessel dimensions were constrained by the physical “keep-out zones” of the optical bench within the gondola enclosure. However, even choosing the maximum possible dimensions for the electronics enclosure would require deliberate placement of all interior components. Figure 5 shows a CAD depiction of the pressure vessel mounted to the optics bench it controls.

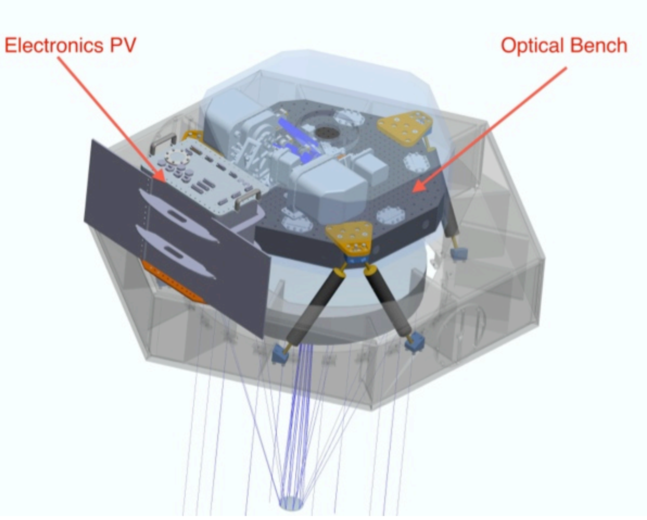


Figure 5: The UVVIS electronics pressure vessel was mounted on the optical bench, attached to the back of the telescope.

Even utilizing the largest volume possible, fitting all the electronic components within such a small pressure vessel was an ongoing struggle. Wire lengths and routes had to be carefully chosen, as excess cabling would keep the electronics stack from fitting into its enclosure. Figure 6 shows the CDH/EPS stack, in a flight configuration on its mounting rack. Figure 7 shows the tight fit of the electronics inside the pressure vessel.

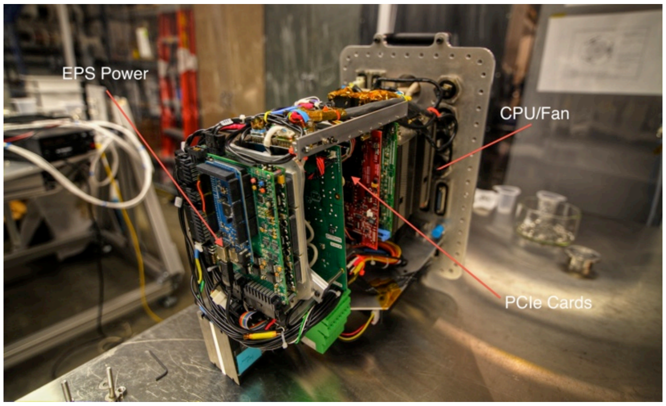


Figure 6: Prior to integration, this is a view of the compact brick of electronics that make up the UVVIS flight computer and power system.

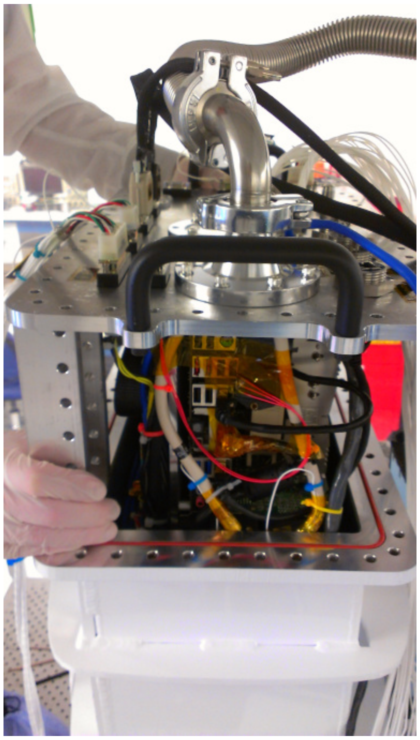


Figure 7: The electronics barely squeeze into the pressure vessel.

Figure 8 shows a custom rail that was built to provide mechanical stability and electronic isolation for the peripheral PCIe cards used by the CDH system. The mounting for these components were intended for used in a benign desktop environment, not one which would be attached to a moving telescope. While securing PCIe cards may seem trivial, slight misalignment of these COTS cards led to many difficult-to-debug issues, such as intermittent functionality and diminished communication. Iteration was required to find a physical configuration that would securely fasten these cards despite their dynamic orientation with respect to gravity.

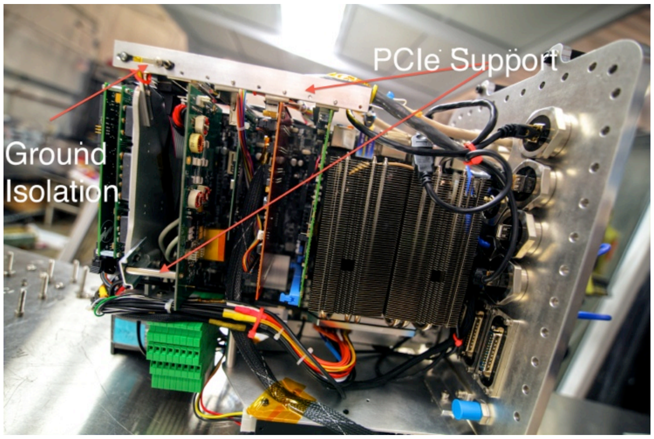


Figure 8: The PCIe cards required a custom mechanical support to keep a sound electrical connection regardless of the orientation of the electronics box.

The prevalent challenge faced with a COTS-based flight computer design, particularly one needing high CPU performance, is thermal mitigation. For the UVVIS electronics box, it was determined early that the entire CDH system would be housed in a 1 atm pressure vessel to minimize the unknowns in exposing COTS computer hardware to near-space conditions. Fans inside the box circulate air throughout and transfer heat to the internal aluminum structure. Attached to the outside of the box are two 16.5” x 14” radiators to release heat into the space environment.

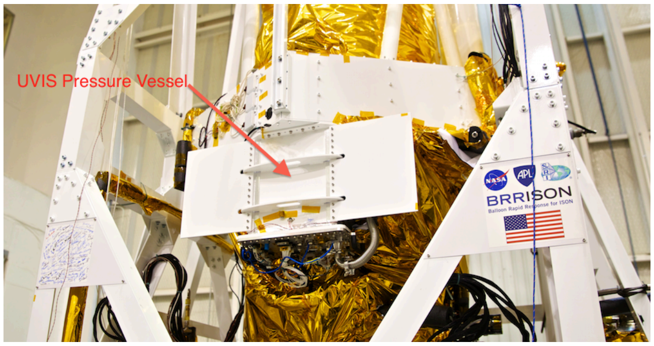


Figure 9: The UVVIS electronics box is mounted to the side of the BRRISON telescope with two radiators installed.

The environment at flight altitude (120K ft) is reasonably well known, but how the CDH computer components would behave inside a tightly packed box dissipating a total of 150 watts was a different story. In most space systems components have hard thermal requirements, however, dealing with the “recommended operational temperatures” indicated by most COTS specifications made the problem even more difficult. Determining whether the assembled computer would thermally survive flight, or even ground testing, for that matter, proved to be a tremendous undertaking. The environment inside the pressure vessel was impossible to model with any considerable level of confidence. The tight proximity of components and the varying nature of heat loads complicated matters. For example, the main CPU alone was capable of outputting over 75 Watts of heat during high-use scenarios, but while idling, it would produce less than one-third of that. Other components had similarly large swings in heat production, depending on their usage. With limited time to perform adequate convection modeling, SwRI decided that simply testing the box in an altitude chamber was the most time and cost efficient method for proving survivability.

In the end, the system did indeed stay within its thermal range during all ground testing and during flight. Flight conditions were the most benign that the system faced. Indeed, ground testing thermally stressed the system the most, with CPU and ambient temperatures often exceeding 85° and 60° Celsius, respectively. However, it is worthy to note that the thermal performance of the CDH pressure vessel was the single most discussed element of the entire system. Far more attention was spent on the possible thermal performance and mitigation strategies than on hardware or flight software. In terms of pure resources, the manpower spent just discussing the issue cost orders of magnitude more than the entire set of flight hardware.

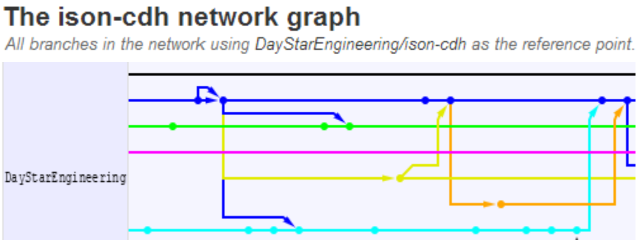
In all, the choice to use COTS computer hardware was beneficial in that it provided rapid development and modularity. However, the resulting CDH system was, admittedly, not as efficient as a custom solution might have been, especially in volume and power. This was an allowable concession, as short balloon missions have both of these in relative abundance.

Flight Software

The UVVIS flight software needed to be robust, reliable, and quick to develop. For this purpose, a Linux operating system was chosen. Linux offers full featured multi-threading and inter-process communication capabilities. This was particularly useful in serving the wide array of needs required by the UVVIS bench.

The flight CDH system and the development units all ran on the latest version of Ubuntu, version 12.10 in February 2013. Ubuntu was chosen due to the extraordinary amount of online support. The flight computer runs a minimalistic Ubuntu version, free of unnecessary bloat like a GUI, while the build computers run full versions. Using nearly identical software and hardware configurations on both the build and flight computers proved an invaluable resource. Peripherals could be plugged into either the build or flight units for testing. No cross compiling was necessary, and a fair amount of prototyping could be done without even copying binaries to the flight computer. This saved an enormous amount of development time throughout the project.

The first step SwRI took in software development was to construct a Git version control based workflow. This was unequivocally the most powerful tool employed in the entire project. The branching techniques, commit strategies, and a team-based workflow that Git offers accelerated development. Two or more team members would often be working simultaneously on the flight software. Each worked on their own branch, testing their changes on different components of the flight machine. At a day's close, branches would be merged and all software changes seamlessly integrated into the master code set. The flexibility of this branching strategy is visualized in Figure 10. Git effortlessly merges hundreds of lines of code within a single file, and allowed SwRI to concurrently manage dozens of branches in various states of completion.


The ison-cdh network graph

All branches in the network using DayStarEngineering/ison-cdh as the reference point.

Figure 10: Git version control offers lightweight branching and merging that proved invaluable to the timely development of the UVVIS flight software.

The code set was duplicated for each user on multiple computers and backed up on the GitHub servers for redundancy and analysis. GitHub offers a variety of useful metrics, such as the amount of code written by each developer, at what time that code was submitted, and visual diffs of each Git *commit*. Figure 11 shows the *commit history*, a set of software patches, for a single user on the UVVIS project.

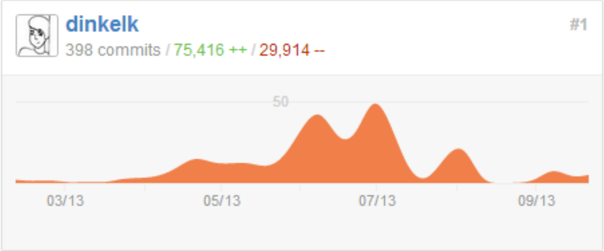


Figure 11: GitHub.com provides useful visualizations to keep track of each developer's contributions to a project. Kevin Dinkel's 398 contributions are graphed above, showing that he wrote over 75,000 lines of new software for the UVVIS CDH system in just 5 months.

Flight Software Challenges

Without senior software engineers on the team, the open source community became the legacy personnel. SwRI took full advantage of the internet community and their wealth of experience on web metropolises such as GitHub and StackExchange. Whenever the choice was presented, open-source software tools were used. The only closed-source libraries used were those provided by the manufacturers of specific hardware, such as the scientific and guide cameras. Not surprisingly, it was with these proprietary systems that the SwRI developers spent most of their time debugging, calling manufacturers, and working out problems. The difference in quality between the open and closed parts of the UVVIS system was astonishing. Getting the camera SDKs up and running, alone, took 8 weeks of effort, including daily emails and weekly teleconferences with each company. During this process SwRI uncovered dozens of bugs in both camera SDKs. The camera manufacturers design their cameras as lab instruments, whose primary interface is through Microsoft Windows. Linux SDKs were provided with each, but getting drivers and hardware to work with Linux was much harder than the manufacturers indicated. Making minute tweaks in BIOS settings and dealing with proprietary software designed for decade old hardware, caused many issues that were difficult to diagnose and remedy. Throughout the process, posting to online forums and questioning the community proved equally, and in some cases more, useful than speaking with the manufacturers directly. Overall, when constructing custom flight software from scratch, SwRI found that leveraging open source solutions was much simpler and generated higher quality code than using the proprietary SDKs provided by the camera manufacturers.

Because the hardware components used for the mission were not designed specifically for SwRI's intended engineering use, there were several additional hurdles to overcome. For instance, COTS computer hardware is usually designed for interaction with a desktop user. This can be especially troublesome if there is no way to bypass user interactivity, a requirement for autonomous systems on a balloon. As an example, the RAID hardware chosen for redundant data storage was abandoned after it was discovered that the hardware could not autonomously boot into the operating system after a drive failure. Instead, it would stall the computer operation prior to boot, waiting for user input. Unfortunately, this meant that the benefits of redundancy through RAID during flight were undermined by the fact that the operating system would no longer boot after drive failure. Unfortunately, this was the case for every consumer RAID manufacturer SwRI spoke with. For this reason SwRI needed to implement custom data redundancy software.

Similar pitfalls were found when booting the computer. After new hardware was installed or power was cut unexpectedly from the system, boot-blocking BIOS splash screens appeared. While most of these could be disabled in the BIOS, some were remedied by con-ops plans to reboot the computer multiple times after a boot failure.

In all, consumer hardware is not rendered totally inoperable by off-nominal behaviors, but the requirement for human-interaction can present unexpected challenges. Budgeting this kind of "experimental" testing time into a COTS hardware project is an absolute must.

2. EPS SUBSYSTEM

The UVVIS power system is designed to power the flight computer, cameras, mechanisms, and thermal control hardware on the optics bench. In addition to providing DC voltages for each of these loads, the EPS is responsible for switching and sensing each voltage rail. Finally, to operate real-time on flight, the EPS is required to receive commands from and send data to the flight computer.

One of the goals in the design process was to produce a modular and reusable power system. At a high level, this meant providing a variety of voltages, as well as a degree of configurability. The EPS was given sufficient margin to power many extra loads. It was also designed in a vertical stack, so new boards could be added as needed. The system components and capabilities are discussed in the following sections, with the focus on rapid design and modularity.

Power Control PCBs

Two PCBs make up the core of the UVVIS EPS. The first is the Main Power Board (Figure 12), which is centered on several COTS DC converters. These converters provide 12V for the Zyla and 24V for the ProEM, fold mirror, filter wheel and heater controller. Two ±15V rails are intended for the fine steering mirror. Finally, 5V and 3.3V buck converters power all the onboard chips, and can be extended to additional loads.

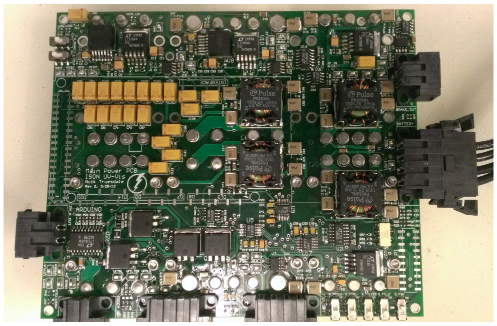


Figure 12: The Main Power Board provides the conversion, switching, and sensing for all of the EPS loads except the flight computer.

Due to the compressed design time, converter selection was limited to rugged COTS parts. The 12V and 24V converters are Vicor Mini bricks, with Vicor MicroRAM ripple attenuators on the outputs. The ±15V are similarly Vicor Micro bricks. These DC-DC bricks offer several advantages: the outputs are configurable, power ratings range from 50W to 350W, and the output voltages meet common ripple requirements. The bricks mount via sockets, and can be reconfigured without redesigning the PCB. This means the Main Power Board can be reconfigured to accommodate nearly all common electronics.

The 5V and 3.3V converters are predesigned LMZ22005 modules from Texas Instruments that also have significant power overhead. This overhead can be applied to any low-level chips or boards that might be added to the system.

The second core PCB is the Computer Power Board. This is a custom ATX computer power supply that can run off of a DC input between 18V and 36V. The input is converted by a COTS VME-550 regulator board, seen in Figure 14. The outputs of the VME-550 are 12V and 5V rails with 224W capacity each, and 3.3V and -12V rails with 112W. This VME card did most of the legwork required for conditioning signals for a COTS computer.

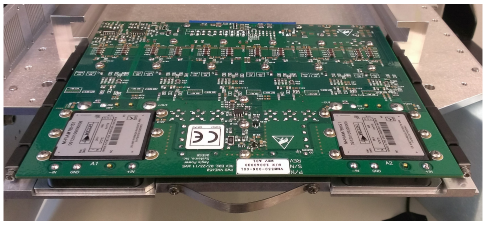


Figure 13: A VME-550 from Aegis produces 12V, 5V, 3.3V and -12V for the ATX motherboard.

Following the VME-550, the Computer Power Board (Figure 14) controls the startup and shutdown timing for the computer. The board meets all of the ATX specifications for voltage and current timing. Standard 24-pin and 8-pin power cables supply the motherboard, and a header extends the motherboard power switches and LEDs. This allows a user to turn on the computer using a pushbutton, and monitor the computer's power state.

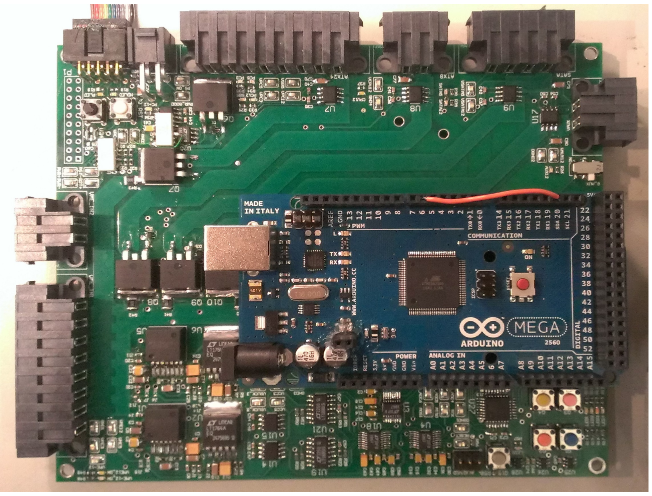


Figure 14: The Computer Power Board controls the startup and shutdown of the UVVIS computer.

The Main Power Board and Computer Power Board provide a complete power solution for the UVVIS payload, as well as most other balloon payloads. Together, the two boards allow a user to power numerous instruments, and control them with any ATX motherboard/processor and standard computer peripherals. The use of COTS power components allowed this to happen in only five months, while maintaining modularity.

Arduino Microcontroller

The only feature not provided by the two core power PCBs is logic. Because of the rapid design timeline and a lack of legacy solutions, it was not feasible to design custom embedded logic. A COTS solution was desired that could control all of the boards at once. The Arduino Mega 2560 pictured in Figure 15 provides the EPS stack with 69 GPIOs, enough to control all of the instruments and the flight computer simultaneously. The Arduino is ideal for rapid design, because it includes a USB interface that supports programming. Using the Arduino C code libraries, one can write and deploy low-level logic code to power the EPS with little custom software, but with the speed and ease of a higher-level language. All software is open-source and supported by large friendly community of users and developers. This allowed the EPS to be tested with ease on the ground and become quickly configured for flight.

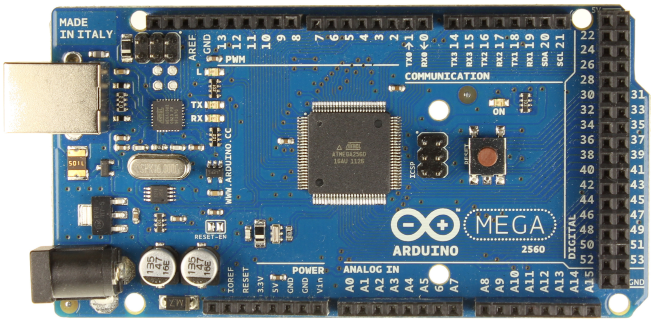


Figure 15: The Arduino Mega 2560 microcontroller board is an inexpensive, well supported component that offers a USB programming interface and well documented code library.

The Arduino footprint, shown in Figure 16, was built into each PCB in the UVVIS EPS. This offered flexibility during test, because the microcontroller could be mounted to any set of boards. It is also what allows the boards themselves to stack together, as is discussed in the next section.

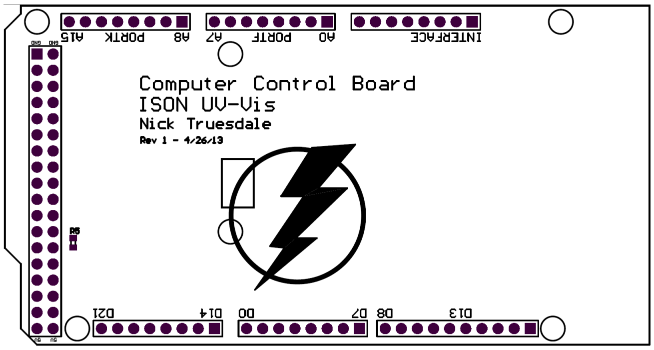


Figure 16: The footprint for the Arduino is on each EPS board, allowing the logic to be integrated for any combination of boards.

UVVIS EPS Stack

The UVVIS was designed not only to be reusable for future missions, but also to be compact and modular within the scope of the BRRISON mission. To achieve this, the boards are arranged in a vertical stack, as shown in Figure 17. This architecture has many advantages. First, it conserves space; the EPS as flown only takes up a 5” x 6” x 3” volume. Second, the stack allows new boards to be added using a common vertical bus; this capability is discussed in the next section. Finally, the stack is modular, allowing boards to be stacked in different orders. This is particularly useful for testing, where a board might need to be removed or tested alone. The needed boards can be combined with the Arduino in any test configuration.

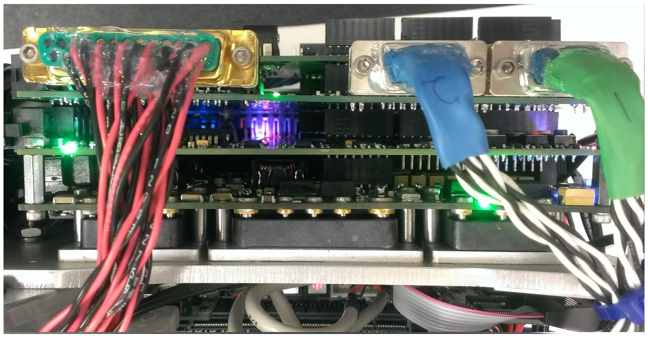


Figure 17: The EPS system stacks vertically, with the power bricks on bottom mounted to an aluminum heatsink and the Arduino on top.

Figure 17 shows the EPS stack as it was flown. The Main Power Board is on the bottom, because its DC-DC bricks require an aluminum heatsink. The Computer Power Board mount next, and the Arduino sits on top. The third board in the stack was added during integration: this extension is discussed in the next section.

Extensibility of the EPS

As noted previously, one of the biggest advantages of the UVVIS EPS is that it is extensible. The stack architecture permits any number of boards to be added. This was utilized in the final weeks of integration, when the thermal system’s relays needed to be replaced with a more compact switching system. The new system was integrated on a third PCB, called the Heater Control Board. This board holds eight MOSFET switches, and can be controlled from the Arduino or an external PID controller. The advantage of having this capability as part of the EPS is that the Arduino can now sense heater voltage and current, and has the ability to shut off heaters in the case of an emergency.

The Heater Board is shown mounted atop the EPS stack in Figure 18. In addition to the switching circuitry, the board holds an LCD screen at the bottom, and can control a second one remotely. This, once more, showcases the usefulness of the stack design; because the logic and power already existed, adding complex new electronics became as easy as laying the footprint on a new PCB. The LCD screens can be used to output the state of the EPS, providing a simple user interface for testing. This is especially useful when the stack is integrated, because the screen displays information that would otherwise be hidden under several boards.

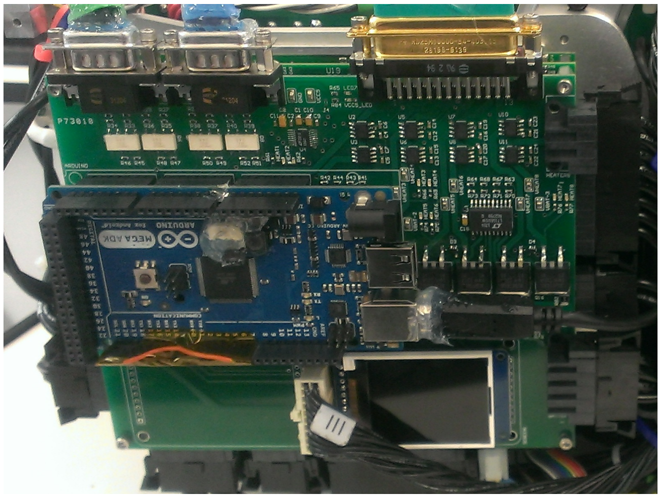


Figure 18: The Heater Control Board, shown here integrated, adds thermal control capability to the EPS.

4. CONCLUSION

The BRRISON project truly required a “rapid response” in order to achieve mission success. With minimal legacy experience or design, SwRI was able to build a fully functional balloon-borne power and control system in only 5 months with 4 part-time engineers. This could only be accomplished with aid of COTS hardware and open source software. While aerospace projects tend to reject consumer grade and open source tools for mission critical components due to their lack of flight heritage, SwRI found that with diligent research and testing, their use can save enormous amounts of development time and costs. Moreover, the wealth of support and thriving community that comes with these tools makes debugging and integration of COTS/open source systems much simpler than they otherwise would be. Throughout the design, integration, testing, and finally flight phases of this project, the SwRI team never felt disadvantaged by their choices in hardware and software. The final system met all expectations, and proved COTS/open-source solutions can perform in aerospace systems. In the end, SwRI designed a modular, extensible CDH/EPS system that can readily be reused and modified for future balloon missions.

REFERENCES

[1] http://brrison.jhuapl.edu/ballooncraft/instruments.php

[2] E. Young, T. Kremic, C. Hibbitts, J. Dankanich, R. Woodruff, S. Osterman, M. Davis, K. Dinkel, Z. Dischner, N. Truesdale, J. Diller. “Addressing the Goals in the Planetary Science Decadal Survey from a Balloon-borne Platform” 21st ESA Symposium 2013. June 9, 2013

BIOGRAPHY

Zach Dischner *graduated with a BS in Aerospace Engineering in 2012 from the University of Colorado at Boulder. Currently, he is a graduate student pursuing a MS in Aerospace Engineering Systems. He is a graduate data systems technician for the Laboratory for Atmospheric and Space Physics (LASP). Zach is a fervent outdoorsman, whose main interests include skiing, kiteboarding, rock climbing, as well as freelance photography.*

Kevin Dinkel *is an Aerospace Engineering graduate student and a software engineer at the University of Colorado at Boulder. He has been a lead software systems engineer for 5 high-altitude balloon payloads, including BRRISON, DayStar, and BOWSER. He has also developed flight software for the DANDE nano-satellite and for an experimental payload aboard the Surrey Orbital Test Bed. Kevin previously worked on ground and thermal systems at Lockheed Martin Space Systems and BITSystems Co., and currently works as a software and systems engineer at the Southwest Research Institute (SwRI) and Laboratory for Atmospheric and Space Physics (LASP) in Boulder, CO.*

Nick Truesdale *is pursuing an M.S. in Aerospace Engineering and a B.S. in Applied Mathematics at the University of Colorado. He was the Electrical Systems Lead for the BOWSER and DayStar balloon payloads, and the Power Lead for the ALL-STAR CubeSat. Through his work on these projects, Nick has gained experience in power systems, microcontrollers, PCB design and layout, and general electrical testing. He is also a skilled programmer, and combines his wide array of skills in system design. Nick has worked with Lockheed Martin, MIT Lincoln Laboratory, and is now supplements his studies as an independent contractor in electrical and systems engineering. He is currently managing the OTCCD and ISON projects for Southwest Research Institute*

Jed Diller *graduated with a B.S. in Aerospace Engineering from the University of Colorado at Boulder in 2012. He is lead engineer on the SwRI Solar Instrument Pointing Platform (SSIPP) and controls lead on the UVVis Instrument (BRRISON payload) for the Southwest Research Institute (SwRI) in Boulder, CO. His former professional work includes software engineering for geophysical software companies and the University Corporation for Atmospheric Research*