Command and Data Handling

Introduction
There are two primary trends in small spacecraft command and data handling (C&DH). Incorporating small spacecraft, especially CubeSats, into more complex science and technology applications in LEO and deep space or interplanetary missions, requires increased system reliability and performance. In the case of the smaller spacecraft, these objectives are complicated by the use of highly integrated systems and the need for power and mass efficiency.

At the other end of the spectrum, low-cost, easy-to-develop systems that take advantage of open source software and hardware are providing an easy entry into space systems development, especially for hobbyists or those who lack specific spacecraft expertise.

The author would like to highlight that the presented tables are not intended to be exhaustive but to provide an overview of current state-of-the-art technologies and their development status for this small spacecraft subsystem. There was no intention of mentioning certain companies and omitting others based on their technologies.

State of the Art
Since the publication of the first edition of this report, several CubeSats using commercial-off-the-shelf (COTS) components and integrated systems have successfully flown in the LEO environment, over short mission durations of typically less than one year.

A variety of C&DH developments for CubeSats have resulted from in-house development, from new companies that specialize in CubeSat avionics, and from established companies who provide spacecraft avionics for the space industry in general. Presently there are a number of commercial vendors who offer highly integrated systems that contain the on-board computer, memory, electrical power system (EPS), and the ability to support a variety of input & output (I/O) for the CubeSat class of small spacecraft.

In anticipation of extended durations in LEO and deep space missions, vendors are incorporating radiation hardened or radiation tolerant designs in their CubeSat avionics packages.

Form Factor
The CompactPCI and PC/104 form factors continue to be the industry standard electronics bus systems with multiple vendors offering components that can be readily integrated into space rated systems. Overall form factors should fit within the standard CubeSat dimension of less than 10 x 10 cm.

The PC/104 board dimension continues to be the baseline for CubeSat configurations. Many vendors have adopted the use of stackable “daughter” or “mezzanine” boards in order to simplify connections between subsystem elements and payloads, as well as to accommodate advances in technologies that maintain compatibility with existing designs. A few vendors provide a modular package which allows users to select from a variety of computational processors.

On-Board Computing
Microcontrollers and FPGAs
Small spacecraft, and especially CubeSat developers, continue to use microcontrollers and field programmable gate arrays (FPGAs) to support a variety of different processor cores. FPGAs have
space heritage and integrate easily with peripherals, on-chip memories and improved power performance—all factors that influence the choice of on-board computing at present. See Table 8-1 for the current state-of-the-art for highly integrated on-board computing systems for small spacecraft.

<table>
<thead>
<tr>
<th>Product</th>
<th>Manufacturer</th>
<th>Processor</th>
<th>TRL Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nanomind A712D</td>
<td>GomSpace</td>
<td>ARM7</td>
<td>9</td>
</tr>
<tr>
<td>ISIS OBC</td>
<td>ISIS</td>
<td>ARM9</td>
<td>9</td>
</tr>
<tr>
<td>Pluggable Socketed Processor Module</td>
<td>Pumpkin</td>
<td>C8051F120, PIC24F256110, PIC 24F256GB210, MSP430F1612, MSP430F1611, MSP4302618</td>
<td>9</td>
</tr>
<tr>
<td>MODAS</td>
<td>Utah State University</td>
<td>TI320C6713DSP</td>
<td>9</td>
</tr>
<tr>
<td>RAD750</td>
<td>BAE</td>
<td>ATMEL AT91SAM9G20</td>
<td>9</td>
</tr>
<tr>
<td>Intrepid</td>
<td>Tyvak</td>
<td>ATMEL ATmega328P</td>
<td>9</td>
</tr>
<tr>
<td>Q5, Q6, Q7</td>
<td>Xiphos</td>
<td>PowerPC 405, Xilinx Spartan-6, Actel ProASIC3 Control FPGA, Xilinx Zynq 7020 ARM dual core Cortex A9, Actel ProASIC3 Control FPGA</td>
<td>9</td>
</tr>
<tr>
<td>ArduSat</td>
<td>NanoSatisfi</td>
<td>ATMEL ATmega328P</td>
<td>9</td>
</tr>
<tr>
<td>Medusa, HPD, RCC5, Athena-2 SBC, FMC Gen2/3, AIP</td>
<td>SEAKR</td>
<td>2450 DMIPS, PowerPC e500 core, Xilinx Virtex 5 FPGA, Virtex-5 FX-130T FPGAs, CCSDS, Leon Processor card, RCC4-LX160 &amp; RCC4-LX200</td>
<td>9</td>
</tr>
</tbody>
</table>

Many power efficient microcontrollers used in CubeSats feature ARM processors and a variety of on-chip peripherals, such as universal serial buses (USB), controller area networks (CAN), as well as I2C interfaces and serial peripheral interfaces (SPI). There has also been an increase in the number of microcontrollers that use programmable flash memory.

System developers are gravitating towards ready-to-use hardware and software development platforms that can provide seamless migration to higher performance architectures. As with non-space applications, there is a reluctance to change controller architectures due to the cost of retraining and code migration. Following the lead of microcontrollers and FPGA vendors, CubeSat avionics providers are working towards providing simplified tool sets and cost effective basic evaluation boards.
Smartphone-based Processing

NASA’s PhoneSat-1.0 and SSTL’s STRAND-1 flew CubeSats that used Google Nexus One smartphones as the central processor, further demonstrating COTS hardware. Smartphones exploit a large market with a fast design cycle, and incorporate several key features that are used in spacecraft, such as cameras, radio links, accelerometers and high-performance computer processors. The smartphone cores used on those early spacecraft were based upon the Qualcomm Snapdragon system on chip (SoC) with a 1 GHz Scorpion processor running the Android operating system. PhoneSat-1.0 simply flew the phone in a CubeSat chassis along with a battery pack for power and a UHF beacon radio.

The hobbyist market that has subsequently emerged from smartphone app development experienced the same I/O bottlenecks and mounting problems observed by these smartphone spacecraft. Consequently, a range of low-power microprocessors are now available, although these are still based on ARM and often run Android, but provide better modularity. No smartphone-based CubeSat avionics kits are commercially available at this time.

Open Source Platforms

A number of open source hardware platforms hold promise for small spacecraft systems. Arduino boards consist of a microcontroller with complementary hardware circuits, called shields. The Arduino platform uses Atmel microcontrollers, therefore developers can exploit Atmel’s development environment to write software. The ArduSat spacecraft used the Arduino platform and successfully engaged the public to raise funding on Kickstarter. BeagleBone has also emerged as a popular open source hardware platform. BeagleBone contains an ARM processor and supports OpenCV, a powerful open source machine vision software tool that could be used for imaging applications. BeagleSat is an open source CubeSat platform based on the BeagleBone embedded development board. It provides a framework and tool set for designing a CubeSat from the ground up, while expanding the CubeSat community and bringing space to a broader audience. Raspberry Pi is another high-performance open source hardware platform capable of handling imaging, and potentially, high-speed communication applications (Wooster, Boswell, Stakem, & Cowan-Sharp, 2007).

Finally, Intel has entered the market with their Edison system. The dual-core x86-64 SoC was targeted at “Internet of Things” applications, but the Edison has proven to be very well suited for advanced CubeSat development--a novel use that Intel has embraced.

Arduino has become known for being beginner friendly, and making the world of microcontrollers more approachable for software designers. Though it presents a relatively familiar set of API to developers, it does not run its own operating system. On the other hand, the BeagleBone Black, Raspberry Pi, and Intel Edison are full-featured embedded Linux systems running Angstrom, Raspbian, and Yocto Linux kernels out of the box respectively. This broadens the range of developer tool options, from web-based interfaces to Android and Python environments. Not only does this further ease the learning curve for novice developers, but it allows the full power of a Linux system to be harnessed in computation tasks.

Memory and Electronic Components

The range of on-board memory for small spacecraft is wide, typically starting around 32 kB and increasing with available technology. For C&DH functions, on board memory requires high reliability. A variety of different memory technologies have been developed for specific traits, including static random access memory (SRAM), dynamic RAM (DRAM), flash memory (a type of electrically erasable, programmable, read-only memory), magnetoresistive RAM (MRAM),
ferro-electric RAM (FERAM), chalcogenide RAM (CRAM) and phase change memory (PCM). SRAM is typically used due to price and availability. A chart comparing the various memory types and their performance is shown in Table 8-2.

<table>
<thead>
<tr>
<th>Feature</th>
<th>SRAM</th>
<th>DRAM</th>
<th>Flash</th>
<th>MRAM</th>
<th>FERAM</th>
<th>CRAM/PCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-volatile</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Operating Voltage, ±10%</td>
<td>3.3 – 5 V</td>
<td>3.3 V</td>
<td>3.3 &amp; 5 V</td>
<td>3.3 V</td>
<td>3.3 V</td>
<td>3.3 V</td>
</tr>
<tr>
<td>Organization (bits/die)</td>
<td>512k x 8</td>
<td>16M x 8</td>
<td>16M x 8; 32M x 8</td>
<td>128k x 8</td>
<td>16k x 8</td>
<td></td>
</tr>
<tr>
<td>Data Retention (@ 70°C)</td>
<td>N/A</td>
<td>N/A</td>
<td>10 years</td>
<td>10 years</td>
<td>10 years</td>
<td>10 years</td>
</tr>
<tr>
<td>Endurance (Erase/Write cycles)</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>10⁶</td>
<td>1013</td>
<td>1013</td>
<td>1013</td>
</tr>
<tr>
<td>Access Time</td>
<td>10 ns</td>
<td>25 ns</td>
<td>50 ns after page ready; 200 s write; 2 ms erase</td>
<td>300 ns</td>
<td>300 ns</td>
<td>100 ns</td>
</tr>
<tr>
<td>Radiation (TID)</td>
<td>1 Mrad</td>
<td>50 krad</td>
<td>30 krad</td>
<td>1 Mrad</td>
<td>1 Mrad</td>
<td>1 Mrad</td>
</tr>
<tr>
<td>SEU rate (relative)</td>
<td>Low-nil</td>
<td>High</td>
<td>Nil (cells); Low (device electronics)</td>
<td>Nil</td>
<td>Nil</td>
<td>Nil</td>
</tr>
<tr>
<td>Temperature Range</td>
<td>Mil-std</td>
<td>Industrial</td>
<td>Commercial</td>
<td>Mil-std</td>
<td>Mil-std</td>
<td>Mil-std</td>
</tr>
<tr>
<td>Power</td>
<td>500 mW</td>
<td>300 mW</td>
<td>30 mW</td>
<td>900 mW</td>
<td>270 mW</td>
<td></td>
</tr>
<tr>
<td>Package</td>
<td>4 MB</td>
<td>128 MB</td>
<td>128 – 256 MB</td>
<td>1 MB</td>
<td>1.5 MB (12 chip package)</td>
<td></td>
</tr>
</tbody>
</table>

There are many manufacturers that provide a variety of electronic components that have high reliability and are space rated (see Table 8-3). A visit to any of their respective websites will show their range of components and subsystems including processors, FPGAs, SRAM, MRAM, bus interfaces, application specific integrated circuits (ASICs), and low voltage differential signaling (LVDS).
Table 8-3: Sample of Highly Integrated On-Board Computing System Manufacturers

<table>
<thead>
<tr>
<th>Manufacturers</th>
<th>ATMEL</th>
<th>Honeywell</th>
<th>STMicroelectronics</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAE Systems</td>
<td>Intel</td>
<td>Texas Instruments</td>
<td></td>
</tr>
<tr>
<td>Broadreach</td>
<td>Intersil</td>
<td>3D Plus</td>
<td></td>
</tr>
<tr>
<td>C-MAC Microtechnology</td>
<td>Maxwell Technologies</td>
<td>Xilinx</td>
<td></td>
</tr>
<tr>
<td>Cobham (Aeroflex)</td>
<td>Microsemi (Actel)</td>
<td>Arduino</td>
<td></td>
</tr>
<tr>
<td>Freescale</td>
<td>Space Micro, Inc.</td>
<td>BeagleBone</td>
<td></td>
</tr>
<tr>
<td>AAC Microtech</td>
<td>SEAKR</td>
<td>VORAGO Technologies</td>
<td></td>
</tr>
</tbody>
</table>

**Bus Electrical Interfaces and I/O**

CubeSat class spacecraft continue to use interfaces that are common in the microcontroller or embedded systems world. Highly integrated systems, especially SoC, FPGA and ASICs, will typically provide several interfaces to accommodate a wide range of users and to ease the task of interfacing with peripheral devices and other controllers. Some of the most common interfaces are listed below with a brief description:

- Serial Communication Interfaces (SCI): RS-232, RS-422, RS-485 etc.
- Synchronous Serial Communication Interface: I2C, SPI, SSC and ESSI (Enhanced Synchronous Serial Interface).
- Universal Serial Bus (USB).
- Multi Media Cards (SD Cards, Compact Flash etc.).
- Networks: Ethernet, LonWorks, etc.
- Fieldbuses: CAN-Bus, LIN-Bus, PROFIBUS, etc.
- Timers: PLL(s), Capture/Compare and Time Processing Units.
- Discrete IO: General Purpose Input/Output (GPIO).
- Analog to Digital/Digital to Analog (ADC/DAC).
- Debugging: JTAG, ISP, ICSP, BDM Port, BITP, and DB9 ports.

**Electronic Power Supplies**

A number of developers still design their EPS in-house. This is usually the case when the payload has power control needs and requirements that cannot be met by commercially available suppliers. As the EPS is a critical system for the spacecraft, developers will typically use high reliability or space rated components.

There are several commercially available EPS for the CubeSat platform. These systems provide voltages and regulation typically used in embedded systems such as 3.3 V and 5 V.

These systems also provide an array of features to address end user needs, such as short circuit protection, over current and over/under voltage protection, telemetry, battery charging and monitoring, reset capability and more depending upon the vendor. Many of these systems have
flight heritage and are therefore greater than TRL 6. Please refer to the "Power" section for power supplies for small spacecraft.

On the Horizon

Many C&DH systems will continue to follow trends set for embedded systems. Short duration missions in LEO will continue to take advantage of advances made by industry leaders who provide embedded systems, technologies, and components. In keeping with the low-cost, rapid development theme of the CubeSat-based missions, many COTS solutions are available for spacecraft developers.

Radiation mitigation solutions are being implemented by developers who need to address those concerns for deep space and long duration LEO missions. A brief discussion about those techniques is provided in the "Radiation" subsection below.

Also trending in the CubeSat development arena is the use of open source solutions. A number of C&DH systems being developed are using Linux as their OS. This allows them to take advantage of open source software that has already been developed and tested (Wooster, Boswell, Stakem, & Cowan-Sharp, 2007). NASA has developed open source software to support a number of missions. Others developers are using the open source in its truest sense, providing software libraries and on-line tools to aid in the development of their space systems. A brief discussion on open source is provided in Fitzsimmons, 2012.

Radiation Mitigation and Tolerance Schemes

Deep space and long duration LEO missions will require developers to incorporate radiation mitigation strategies into their respective designs. The CubeSat platform has traditionally used readily available COTS components. Use of COTS parts has allowed for low-cost C&DH development, while also allowing developers to take advantage of state-of-the-art technologies in their designs. Many of the component and system vendors also provide radiation hardened (rad-hard) equivalent devices as well. While there are many commercially available rad-hard components, using these components impacts the overall cost of spacecraft development. In order to keep costs as reasonable as possible, C&DH developers will need to address appropriate use of rad-hard components, along with other radiation mitigation techniques for developing an overall radiation tolerant design as discussed in the following section.

For space applications, radiation can damage electronics in two ways. Total ionizing dose (TID) is the amount of cumulative radiation received. Single event effects (SEE) is the disturbance created by single particles hitting the electronics (Nguyen, 2015). Total dose is measured in krad and can affect transistor performance. Single event upsets (SEU) can affect the logic state of memory. A single event latchup (SEL) can affect the output transistors on CMOS logic, potentially causing a high-current state. This section summarizes techniques used to mitigate system failures caused by radiation effects.

Component Selection

MEMORY

FRAM (Ferroelectric RAM) is a non-volatile random access memory that is persistent like Flash memory. FRAM memory cells are latched using a PZT film structure which is more likely to maintain state during a single event effect than traditional capacitive latches found in RAM (Ball Aerospace & Technologies Corp, 2015), (Henkel, 1996).
**IMAGING**

Charge couple devices (CCD) and complementary metal oxide semiconductors (CMOS) are image sensors that are useful in radiation environments. However, CCD’s are preferred in space applications, while the CMOS detectors are a newer technology for rad hardened image sensors (NASA Goddard Space Flight Center, 2015; Bardoux, Penquer, Gilard, Eoffet, & Auvergne, 2012; Chapman, 2015; Holbert, 2015).

**Protection Circuits**

**WATCHDOG TIMERS**

Watchdog timers are often used to monitor the state of a processor. A watchdog timer is a hardware circuit, external or internal to the processor, which resets the processor when it expires unless refreshed by the processor. If the processor jumps to an erroneous memory location through a single-event upset or a software exception, the watchdog timer resets the processor to restore operations (Mauere, Fraeman, Martin, & D. Roth, 2008).

**COMMUNICATION WATCHDOG TIMER**

A dedicated communication watchdog timer circuit can monitor command and responses to determine if the system is locked up. Such a circuit resets power after a specific number of failed transmissions.

**OVERCURRENT PROTECTION**

Single event latchup (SEL) can cause device failure due to an elevated current state. Hardware and software overcurrent protection can be implemented to watch for elevated current levels and then issue a power reset to the offending circuit. The sampling frequency for software overcurrent protection must be sufficient to detect and reset the subsystem before the elevated current causes permanent damage. For hardware protection, a shunt resistor and bypass diode can be used in conjunction to filter voltage and current spikes for rad hardened devices.

**POWER CONTROL**

Since many components are more prone to radiation effects when powered on, a candidate mitigation strategy is to power off devices when they are not operationally needed.

**Memory Protection**

**ECC MEMORY**

Error-correcting code memory is capable of detecting and correcting bit errors in RAM and FLASH memory. In general, ECC works by storing a checksum for a portion of the memory. This checksum can be used to simply mark a portion of memory unusable and/or correct single-bit errors. The memory controller is responsible for managing the ECC memory during read and write operations (LaBel & al., 1996).

**SOFTWARE EDAC**

Bit errors can be detected and corrected using software. In general, EDAC algorithms use three copies of the memory to detect and correct bit discrepancies. Software routinely “scrubs” the memory, compares each of the three stored memory values, selects the majority value, and corrects the erroneous memory location. Software EDAC can be performed at the bit or byte level. Memory lifetime needs to be considered for software EDAC implementations, since every correction increases the write count to a memory location.
Communication Protection

**SHARED BUS SWITCHING**

Another option is to decouple the clock and data lines so that each peripheral has its own pair. Additional data lines can be used on the master controller. Alternatively, an external FPGA could be used to assign a unique clock/data pair to each peripheral and, optionally, include a method as a way to reconfigure those assignments in flight.

**CRC**

Cyclic redundancy check (CRC) is a common method for detecting memory or communication errors. Parity is a single-bit implementation of a CRC where the bit of summary information is calculated by the XOR of the data to be communicated or stored to memory. For communication channels, a CRC is calculated prior to sending the message, and is appended to the message stream in a known location. When the message is received the CRC is calculated again and compared to the previously-generated CRC appended to the data stream. For memory, the CRC is calculated prior to writing the data to memory. When the data is read out, a new CRC is calculated and compared to the previously generated CRC. CRC’s help detect data corruption but cannot be used to correct the defective data.

**FORWARD ERROR CORRECTION**

Forward Error Correctio (FEC) transmits redundant data to help the receiver recover corrupted data. In its simplest form, FEC could transmit three bits for every bit of data and then vote to restore the original data. More efficient algorithms balance the data overhead with the correction accuracy (Mauere, Fraeman, Martin, & D. Roth, 2008).

**Parallel Processing and Voting**

**TRIPLE MODULAR REDUNDANCY**

Single-event upsets can interrupt discrete logic, including processing. Triple modular redundancy (TMR) is a fault mitigation technique where logic is replicated three times, and the output of the logic is determined by a majority-vote.

**FIRMWARE PROTECTION**

Many spacecraft subsystems include a processor to handle and optimize operations. These processors require firmware which is written into onboard program memory. Like data memory, program memory is also susceptible to single-event upsets and device failure. To counter this issue, a bootloader may be used to check the validity of the firmware and provide a mechanism for uploading new versions. Additionally, multiple copies of the firmware may be stored in memory in case the primary version is corrupt.

**Open Source Spacecraft Software**

Open Source software offers spacecraft developers a way to accelerate software development, improve quality, and leverage lessons learned from prior missions.

**cFS/cFE**

The core Flight System (cFS) and core Executive (cFE) is a set of applications, application framework and runtime environment developed by Goddard Space Flight Center. cFE includes core services like messaging, timekeeping, events, and table-driven commanding and configuration (Fitzsimmons, 2012; NASA Goddard Space Flight Center, 2015).
**COSMOS**

COSMOS is a tool developed by Ball Aerospace that provides a framework for operating and testing an embedded system. The tool includes modules for telemetry display, plotting, scripting, logging, and configuration table management (Ball Aerospace & Technologies Corp, 2015).

**Linux**

Linux is currently supported by several spacecraft avionics providers including Space Micro and Tyvak. Additional software modules are needed for space applications. Such modules may include memory scrubbing, a safe mode controller, watchdog functionality, and other reliability services (NASA Goddard Space Flight Center, 2015).

**Summary**

System level solutions are in demand and a majority of the small spacecraft bus developers use hardware typically employed in the embedded systems and control world. As a result, there are many sources for CubeSat systems, subsystems and components from vendors who provide complete spacecraft bus avionics solutions, which include on-board computing, memory, electronic power supply, and engineering development systems. As CubeSat development and application continues to evolve, there are a wide range of avionics systems and components available to address the needs of the wide range of professional and amateur small spacecraft developers.

Designing and fabricating avionics systems for harsh radiation environments is mitigated by a combination of shielding, derating and controlling operating conditions for cumulative ionization and displacement damage effects that cause gradual degradation in electronic devices. Small spacecraft, especially in the CubeSat class, will need to address impacts of radiation in deep space missions and extended duration missions in LEO. Several processor manufacturers and board level integrators are addressing the need for radiation hardened and radiation tolerant designs. Some board level integrators have also undertaken radiation testing of their integrated systems. Many integrated systems providers, are using radiation hardened processors or FPGAs from manufacturers such as XILINX, ATMEL, Aeroflex.

Open source software and hardware hold a lot of promise for commercial and government spacecraft developers. Making a project open source is the first step. The next step is to socialize the software and encourage developers to not only use, but to contribute back with flight-proven algorithms, software modules, and hardware components.

CubeSats are playing a large role in rapidly developed low-cost missions in space, as they are establishing technology demonstrations and short duration science missions in LEO. NASA and other space agencies are now exploring their application in deep space missions. The CubeSat community will provide innovative solutions to address the reliability requirements necessary for those missions, while attempting to maintain the low-cost approach associated with the platform. Complete avionics packages are available to those who seek an integrated solution. At the other extreme, open source DIY kits are available to those who seek a low-cost way to explore developing their own C&DH system and spacecraft.

For Feedback solicitation, please email: arc-sst-soa@mail.nasa.gov. Please include a business email so someone may contact you further.
References


