SELF-CONFIGURING NETWORK FOR LAUNCH VEHICLE AND SATELLITE AVIONICS

John T. Collins

Microcosm, Inc., 410 Coral Circle, El Segundo, CA 90245

(310) 726-4100 / jcollins@smad.com

L. Jane Hansen and Jon Pollack

HRP Systems, 3878 Carson St., Suite 218, Torrance, CA 90503

(310) 792-4633 / hansen@hrpsystems.com

Abstract

Under an Air Force Phase II SBIR contract, Microcosm is developing a self-configuring avionics network, scheduled for test on a sounding rocket in late 2003 or early 2004. This system is extendable to spacecraft avionics and housekeeping functions, enabling true plug-and-play capability.

1. INTRODUCTION

Imagine creating a distributed hierarchy of data processing, where sensor data conditioning is performed at the sensor based network node and sensor fusion is provided on a different, more available or more centrally located node, further upstream in the data flow. Now focus on an avionics system. Picture a hierarchy (or other combination) of avionics sensors and the associated control law algorithms to incorporate the types of data produced by each of these sensors, either alone or in a fusion capacity. Imagine the same for avionics actuators. Finally, consider the benefits of having each component of the avionics software operating on various nodes of the system, migrating as necessary to optimize processing and bandwidth capabilities. Such a system would capitalize on avionics subsystem component availability, routing the necessary information (no longer just data) to target algorithms, without regard to the hardware host. Not only would this provide for system level fault tolerance, it would also provide a means of creating a more complex algorithmic structure as more processing resources were available within the system. Overall, this type of network and processing concept would provide for graceful degradation of any system, in an autonomous, on-orbit environment.

Figure 1 shows an example of a network implementation with an emphasis on a spacecraft Attitude Control System (ACS). Each node connects through a maximum of 3 (estimate for further review) interconnects with the network, and messages can be routed through different paths from the sender to the receiver. Not all nodes are connected to sensors or actuators; the figure shows a telemetry node, collecting messages and formatting them for a telemetry system, and a service node, connecting the network to a handheld device for test and integration. It also shows a filter node performing a special function, which is not connected to any sensor; it performs part of the function of a traditional ACS/flight computer, such as filtering the data using, for example, a Kalman filter. Another node could be the 'control node', issuing control commands. Several nodes could also employ multiple sensors; for example, the MEMS gyro node would also contain a temperature sensor, which would be used to improve the data accuracy or bias estimate of this particular node.

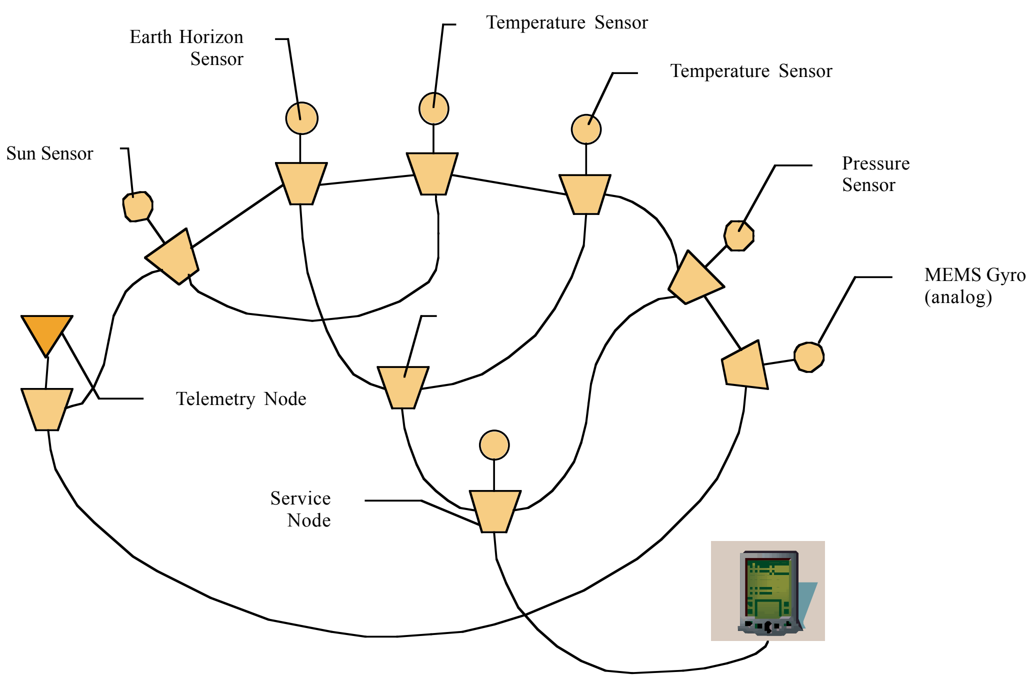

Figure 1. Sample Self-Configuring Network Architecture.

2. SYSTEM DEVELOPMENT

Microcosm is currently under contract to the US Air Force Research Lab to develop a self-configuring, fault-tolerant, plug-and-play networking system for launch vehicle and satellite avionics. This system is meant to be extendable to all other onboard command and control functions, providing a robust, flexible architecture that can accommodate a wide variety of hardware on the network without redesigning the system or updating the software.

In evaluating the need for a Plug-and-Play aspect to avionics system, specifically in terms of a self-configuring avionics network, Microcosm has included the following objectives:

- Reduce development costs by leveraging of COTS hardware/ software and standards
- Increase vehicle reliability by eliminating software/hardware configuration errors and providing built-in redundancy and reconfigurability of flight systems
- Increase vehicle robustness by using standard, protected interfaces
- Increase vehicle performance through realizing lower avionics and associated interconnect mass.

As we develop the system architecture, we need to be certain the communication between various nodes is highly reliable. This implies that there are no contentions on the bus, there can be no single point failure and the communications medium as well as the protocol should be insensitive to noise and ground issues. Thus we have established the need for two types of modules: a passive module and an active module. Specifically, the passive modules will be producers of data. A passive module must be minimally capable of participating in the communication method and respond to a request for data. Then there will be active modules which are the consumers of data. The active modules should have "value added" which implies that the module has enough processing power to assimilate the data and make decisions. A system level architecture will consist of at least two or more active modules and potentially redundant passive modules for each mission critical component of the system.

The specific long-term goals of the modules are that they can be added or removed during operation (hot plugged) and that the other modules will be aware of their existence and function. It is essential for system reliability that no one module is critical to the network operations, however, the mission functionality may be modified based on the availability of various modules. Ideally, if a passive module fails, an active module will attempt to synthesize needed data from secondary producers or extrapolate from previous information. If an active module fails, another active module will assume the responsibilities of that failed unit.

Our protocol is centered around careful and deliberate application of existing protocols and appropriate standards to create an interface protocol configuration that is suitable for unique mission requirements. For example, we will investigate the use of LVDS (low voltage differential signaling) as the electrical protocol. It has the benefit of high data rates and low power usage which is extremely useful in space applications. Additionally, the LVDS protocol generates little EMI noise and is immune to noise interface. It is a low-cost implementation for which radiation hard drivers and hardware are currently available. Next, we will build on the USB-like tree structure of the point-to-point communications links, including enhancements for redundancy. Open Firmware will provide for rigid control of hardware and software interactions in the Plug-and-Play management of device driver configuration control. Finally, the SNMP aspect of the ethernet protocol will be used for top-level command and data delivery management, covering the need for a top-level approach to redundancy and availability.

The passive modules may be built around the 8051architecture. The 8051 architecture has extensive heritage and is available off the shelf. There are multiple suppliers of different 8051 chips so that cost and availability are optimized, while the products are robust and driven by very high-speed clocks which afford a lot of processing power for an 8-bit micro-controller. They have been developed in ruggedized formats and can be integrated with large amounts of mass storage for data collection. Additionally, there are numerous software modules that are currently available for the 8051 architecture which increases availability and provides a means of quickly building up capability with existing, proven components. The software development tools are mature for the 8051 target and the user can quickly become efficient in producing usable software. Another advantage is the number of peripherals that have been developed, tested, and deployed as part of the 8051 architecture. Integrated peripherals such as A/D and D/A converters, pulse-width modulators, temperature sensing, numerous digital I/O pins, and network bus interfaces (UART, I^2C, CAN, etc.) are readily available. Finally, end-of-life issues are mitigated by the current widespread use of this architecture in both commercial and military applications. A university-based small satellite research and development team has created a distributed computing architecture based on this approach that will be flown on the Emerald and QUEST spacecraft. [3] Similarly, the Advanced Instrument Controller, developed at the Air Force Research Lab, is based on the Intel 8031/51 microprocessor. [1]

The active or smart modules may be built around an Intel X86 architecture to provide the needed throughput for the value added processing. As described early in our protocol definition, we will leverage all of the off-the-shelf higher level (TCP/IP) network protocol software components that are readily available for X86 processor. Ruggedized components in a PC-104 format are envisioned for these intermediate, active nodes. Additionally, with the X86 processing in PC-104 format, the available peripherals are wide-open. For example, in the case of mass storage, varying size/quantity and speed of peripherals to meet driving mission requirements will become the key factor rather than the general availability of a mass storage device. While the X86 architecture is more costly than the 8051 architecture, there should be fewer active nodes than passive ones and the added value from additional processing capability as well as the ability to reconfigure the system will outweigh the added costs. Real-time Operating Systems (RTOS) for the X86 architecture are ample.

A final consideration in the selection of a hardware architecture for the passive and active nodes of this self-configuring network is the experience base of the engineering staff at Microcosm. We have had extensive experience with both the 8051 and X86 architectures, using several different RTOSs, and implementing a variety of mission requirements. We have successfully flown an 8051 and an Intel 486 architecture as part of our SR-XM-1 suborbital launch vehicle flight systems avionics suite. The SR-XM-1 flight systems bay is shown in Figure 2. We will fly the self-configuring network architecture hardware as part of the SR-XM-2 suborbital flight.

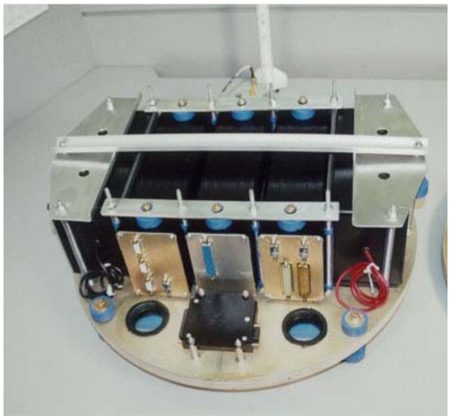

Figure 2. Flight Systems Bay of Scorpius SR-XM-2.

3. System Demonstration

Once we have developed a prototype network architecture, we have a unique opportunity to demonstrate the basis of such an interconnect system in a flight test aboard a Microcosm Scorpius suborbital launch vehicle. The SR-XM-1 is shown in Figure 3. Flight test of such a system will allow for the investigation of pros and cons associated with integration of such a network; evaluation of the ease of adding new components in the network once the hardware has been installed; and demonstration of the overall utility of using this type of network capability in a flight environment. It is not often that such an emerging technology has the opportunity to fly on a near-term test vehicle. However, because Microcosm believes in the inherent benefit of this type of ad hoc network, the technology will be part of the Scorpius® launch vehicle development flight test program.

In addition to the flight test demonstration, we will also create a demonstration of the more advanced features of the network and protocol in a laboratory environment. This will assure a more complete evaluation of the network by performing the testing where the failures of various subsystems and/or components can be controlled, either through simulation or test intervention. When a failure occurs, the performance of the network, and thus the overall system, can be evaluated in terms of system reconfiguration.

In tandem, the flight test and laboratory demonstrations will create a confidence in this network technology that could not be achieved through a paper study or laboratory demonstration alone. When completed, the network hardware and software created by Microcosm will be both flight proven under nominal conditions and laboratory tested under off-nominal conditions. The incorporation of such a network system in operational spacecraft will lead not only to the acceptance of plug-and-play ACS subsystems and components, along with their associated algorithms, but also to the increased reliability and robustness of the ACS system itself. Even if sensors and/or actuators have failed, it will lead to a reduced, but not removed, capability.

4. Future Activities

Microcosm has recently proposed complementary work to the self-configuring network development program that involves creating a standard, plug-and-play spacecraft bus that could be adapted for a wide variety of mission types, and bring down development schedule and cost far below current averages.

Small satellites provide an excellent, near-term test platform for proving new spacecraft technologies. They can also provide a responsive capability for missions that necessitate quick launch, quick operational capability, with an attendant, very short development schedule. Military applications for responsive launch in times of emergency are particularly germane for this technology. Creating a truly modular, plug-and-play spacecraft bus will allow development cycle time to shrink to the order of a few months instead of the usual few years. This will enable rapid testing of new space hardware, as well as creation of highly responsive space systems.

The spacecraft hardware industry has advanced to the point where many components can be considered standard, such as: solar arrays, batteries, power regulators, flight computers, heat pipes, star sensors, Earth sensors, Sun sensors, magnetometers, magnetic torquers, gyros, momentum wheels, thrusters, propellant tanks, data recorders, transmitters, receivers, and antennas. Certain mechanical, electrical, and data interfaces connecting these components to the bus are becoming commonly used and could soon become standard interfaces such as Mil-STD-1553B and ARINC 429/629 in the aircraft industry. Components are also becoming smaller, lighter weight, and lower power as well. The opportunity exists now to create a truly low cost, small, modular satellite bus with standard components and standard interfaces to drastically reduce satellite development schedule and recurring cost. A technology called SpaceFrame is attempting to allow building up of a bus from modular components called Space Frame blocks, which are standardized subsystem modules which can be easily integrated through the use of common, standard interfaces. [2]

Microcosm is pursuing development of a standard, plug-and-play bus under several auspices, one being a recently submitted Phase I SBIR proposal to AFRL on developing this technology and testing it on a suborbital and/or flight within the next 2 to 3 years.

Figure 3. Scorpius SR-XM-1 Suborbital Vehicle.

5. Conclusions

The proposed self-configuring network system will enable decreased development cycle time and cost for satellite and launch vehicle development programs. It will enable hardware swapping with minimal impact to system command and control architecture implementation and software. It will contribute to a truly plug-and-play standard spacecraft bus that can be used for a variety of mission types, with development schedules reduced to months or even weeks.

6. References

[1] Goforth, Todd, Scott R. Cannon, James Lyke, "Space Testing of the Advanced Instrument Controller," 13th Annual AIAA/USU Conference on Small Satellites, Aug. 1999, Paper SSC99-XI-3.

[2] Miller, Jon, Jim Guerrero, David Goldstein, and Tony Robinson, "SpaceFrame: Modular Spacecraft Building Blocks for Plug and Play Spacecraft," 16th Annual AIAA/USE Conference on Small Satellites, August 12–15, 2002, Paper SSC02-III-8.

[3] Palmintier, Bryan, Christopher Kitts, Pascal Stang, Michael Swartwout, "A Distributed Computing Architecture for Small Satellite and Multi-Spacecraft Missions," 16th Annual AIAA/USE Conference on Small Satellites, August 12–15, 2002, Paper SSC02-IV-6.