

Reusable Rack Interface Controller Common Software for Various Science Research Racks on the International Space Station

By George C. Lu

Senior Member of Embedded Software Technical Staff
Boeing Integrated Defense Systems – NASA Systems, Huntsville AL

ABSTRACT

The purpose of the EXPRESS (EXpedite the PROcessing of Experiments to Space Station) rack project is to provide a set of predefined interfaces for scientific payloads which allow rapid integration into a payload rack on International Space Station (ISS). VxWorks® was selected as the operating system for the rack and payload resource controller, primarily based on the proliferation of VME (Versa Module Eurocard) products. These products provide needed flexibility for future hardware upgrades to meet ever-changing science research rack configuration requirements. On the International Space Station, there are multiple science research rack configurations, including:

- Human Research Facility (HRF)
- EXPRESS ARIS (Active Rack Isolation System)
- WOLF (Window Observational Research Facility)
- HHR (Habitat Holding Rack)

The RIC (Rack Interface Controller) connects payloads to the ISS bus architecture for data transfer between the payload and ground control. The RIC is a general purpose embedded computer which supports multiple communication protocols, including fiber optic communication buses, Ethernet buses, EIA-422, Mil-Std-1553 buses, SMPTE (Society Motion Picture Television Engineers)-170M video, and audio interfaces to payloads and the ISS.

As a cost saving and software reliability strategy, the Boeing Payload Software Organization developed reusable common software where appropriate. These reusable modules included a set of low-level driver software interfaces to 1553B, RS232, RS422, Ethernet buses, HRDL (High Rate Data Link), video switch functionality, telemetry processing, and executive software hosted on the RIC computer. These drivers formed the basis for software development of the HRF, EXPRESS, EXPRESS ARIS, WOLF, and HHR RIC executable modules.

The reusable RIC common software has provided extensive benefits, including:

- Significant reduction in development flow time
- Minimal rework and maintenance

- Improved reliability
- Overall reduction in software life cycle cost

Due to the limited number of crew hours available on ISS for science research, operational efficiency is a critical customer concern. The current method of upgrading RIC software is a time consuming process; thus, an improved methodology for uploading RIC software is currently under evaluation.

INTRODUCTION

Under a joint research and development project, NASA, Boeing, and Teledyne Brown Engineering initiated the EXPRESS rack project at Marshall Space Flight Center in early 1994. The pathfinder MSL-1 (Micro-gravity Science Laboratory 1) rack flew on Shuttle Columbia flights STS-83 and STS-94 in 1997. A primary goal of STS-94 was to evaluate facilities associated with the MSL-1 payload. The mission served to bridge the gap between relatively short duration work done on Shuttle Spacelab flights, and the long duration research to be performed on ISS. MSL-1 was activated 14 hours into flight and ran until the fifteenth day of the mission^[1].

MSL-1 utilized the real-time VxWorks® operating system hosted in the multiprocessor RIC environment. The RIC contained a set of Boeing-developed, low-level RS232, RS422, 1553B, Ethernet driver, and executive modules that interface to payloads in MSL-1. Launch schedule and cost requirements were significant drivers in the design of this system.

As a result of successful experiments on STS-94, NASA determined that EXPRESS was capable of adequately supporting on-orbit sub-rack payload operations. The HRF, EXPRESS, EXPRESS ARIS, WOLF, and HHR racks were designed and developed under the auspices of the Marshall Space Flight Center and built by The Boeing Company in Huntsville, Alabama. Two HRF racks, eight EXPRESS racks, one WOLF rack, and two HHR racks are being built for use on the International Space Station. For actual ISS design, Boeing selected similar RS422 and 1553B devices, thus enabling reuse of existing software from the MSL-1

program to the fullest extent possible. Maximizing software reuse was, and continues to be, a major focus of all ISS Payload racks development activity as a mean of ensuring cost-effective, reliable payload software architectures.

HRF RACK

HRF was the first science research rack on ISS designed and built at Marshall Space Flight Center. It provided an on-orbit laboratory enabling life science researchers to study and evaluate the physiological, behavioral, and chemical changes in human beings induced by long-duration space flight^[2]. With the ability to accommodate up to 16 payloads per flight, scientific research performed with the HRF will provide critical data relevant to long term adaptation in micro-gravity environments.

As part of ISS Mission 5A.1, the HRF rack was launched on board Space Shuttle flight STS-102. The HRF rack is shown in Figure 1.

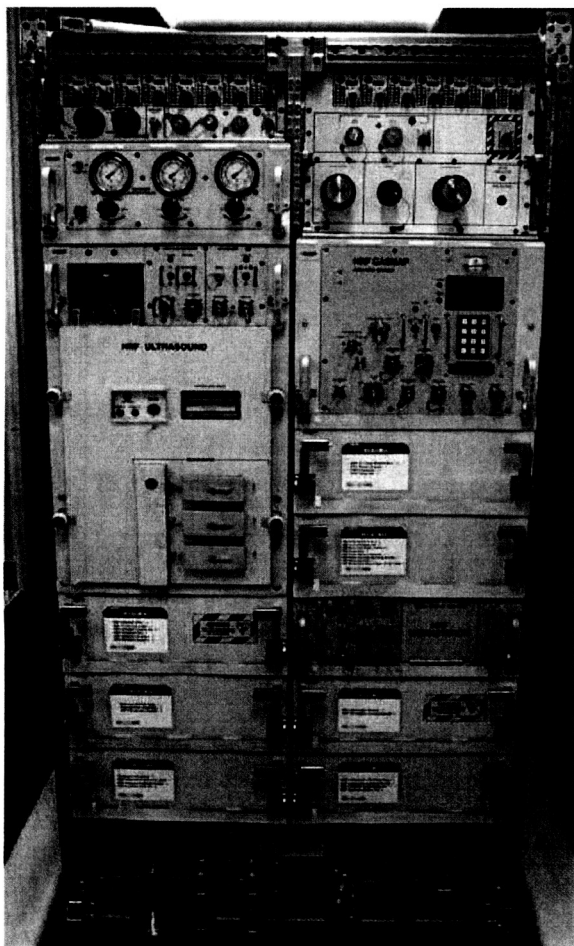


Figure 1: HRF rack houses payloads to investigate the effects of micro-gravity on human physiology^[2]

EXPRESS RACKS

EXPRESS and EXPRESS ARIS are additional variations of science research racks designed for ISS utilization. EXPRESS is a standardized payload rack system designed to transport, store, and support on-board experiments. The EXPRESS system supports science payloads in several disciplines, including biology, chemistry, physics, ecology and medicine^[1].

Even in the quiescent and virtually gravity-free environment of ISS, a minute vibrations or disturbances can perturb (and render useless) sensitive scientific experiments. EXPRESS ARIS acts as a shock absorber between delicate experiments and these disturbances, ensuring that outside influences do not adversely affect costly research results.

EXPRESS accommodates eight mid-deck locker payloads and two standard drawer interface rack payloads. EXPRESS and EXPRESS ARIS were launched on Expedition Two, ISS Mission 6A, STS-101. The EXPRESS rack is shown in Figure 2.

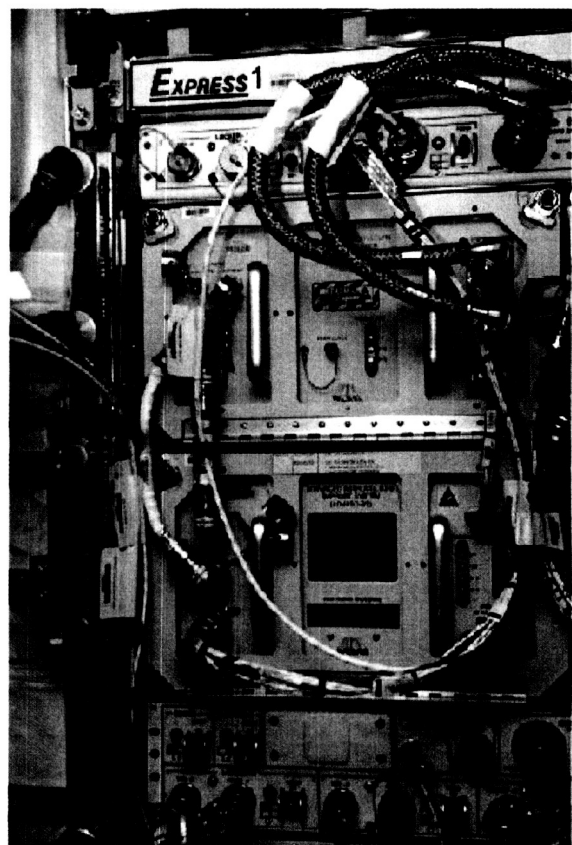


Figure 2: EXPRESS Rack 1 with payloads installed on US Destiny Laboratory Module Expedition Two

WORF RACK

WORF provides capability for advanced earth observation payloads on the International Space Station, and structural accommodations for earth-viewing cameras and sensors. The window in the US Lab is the largest and highest optical-quality window ever flown in space. The Worf rack will launch on STS-114, and subsequently be installed in the US Destiny Laboratory Module at the nadir-pointing research window. Worf is shown in Figure 3.

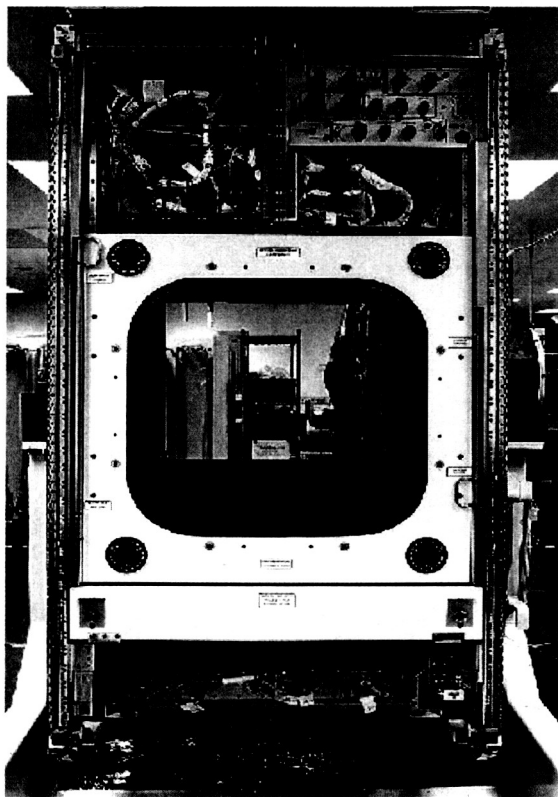


Figure 3: Worf rack prepared for final phase of testing at Marshall Space Flight Center

HHR RACK

HHR is the latest generation of science research rack under development at the Marshall Space Flight Center. The HHR is a host system accommodates the ISS Biological Research Project sub-rack payloads for the study of biological specimens in a low acceleration environment. These specimens include rodent, plant, insect, aquatic, egg, cell, and tissue cultures. The HHR will be launched on ISS Mission UF1, and is shown in Figure 4.

RACK INTERFACE CONTROLLER

The HRF, EXPRESS, Worf, and HHR racks utilize common subsystems for power and command/data handling capability. The RIC serves as the master

controller for each science research rack, providing command and data control between sub-rack payloads, rack subsystems, and the ISS Command and Data Handling (C&DH) System. Three hardware and four software configurations exist for the RIC, with a typical configuration consisting of six VME cards, including:

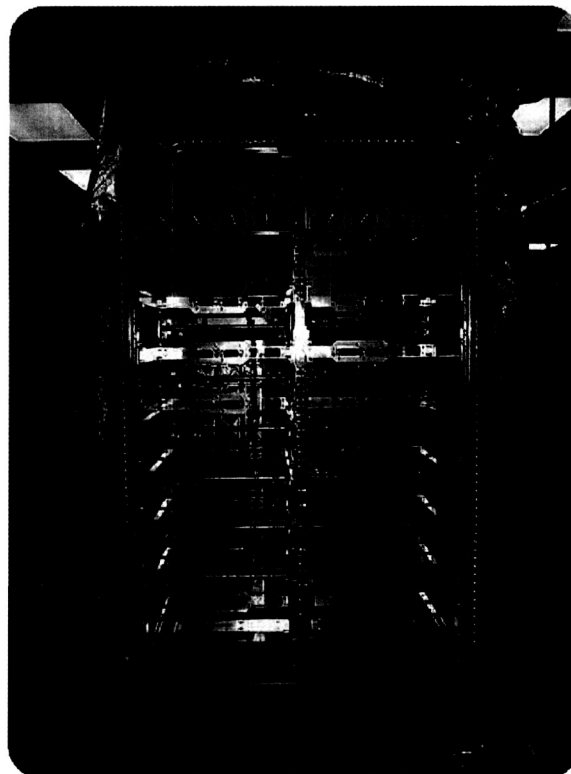


Figure 4: HHR Qualification Rack prepared for power-on testing at Marshall Space Flight Center

- MCC (Main Controller Card) - interface to ISS command and control and ISS LAN 1
- SERC (Serial Card) - interface to RS422 payload
- S1553C (Serial/1553 Card) - interface to RS422 payload, SSPCM (Solid State Power Controller Module), PEHB (Payload Ethernet Hub Bridge), and ISS LAN 2
- HRLC (High Rate Data Link Card) - interface to rack laptop computer, Ethernet payload, EMU (EXPRESS Memory Unit), ISS high-rate telemetry link, and payload video switch interface
- CVIT (Common Video Interface Transmitter) - interface to ISS video downlink through VME back-plane command and control
- VM (Video Multiplexer) - accept payload differential video and distributes to laptop computer, JEM (Japanese Experimental Module) video, and ISS video downlink

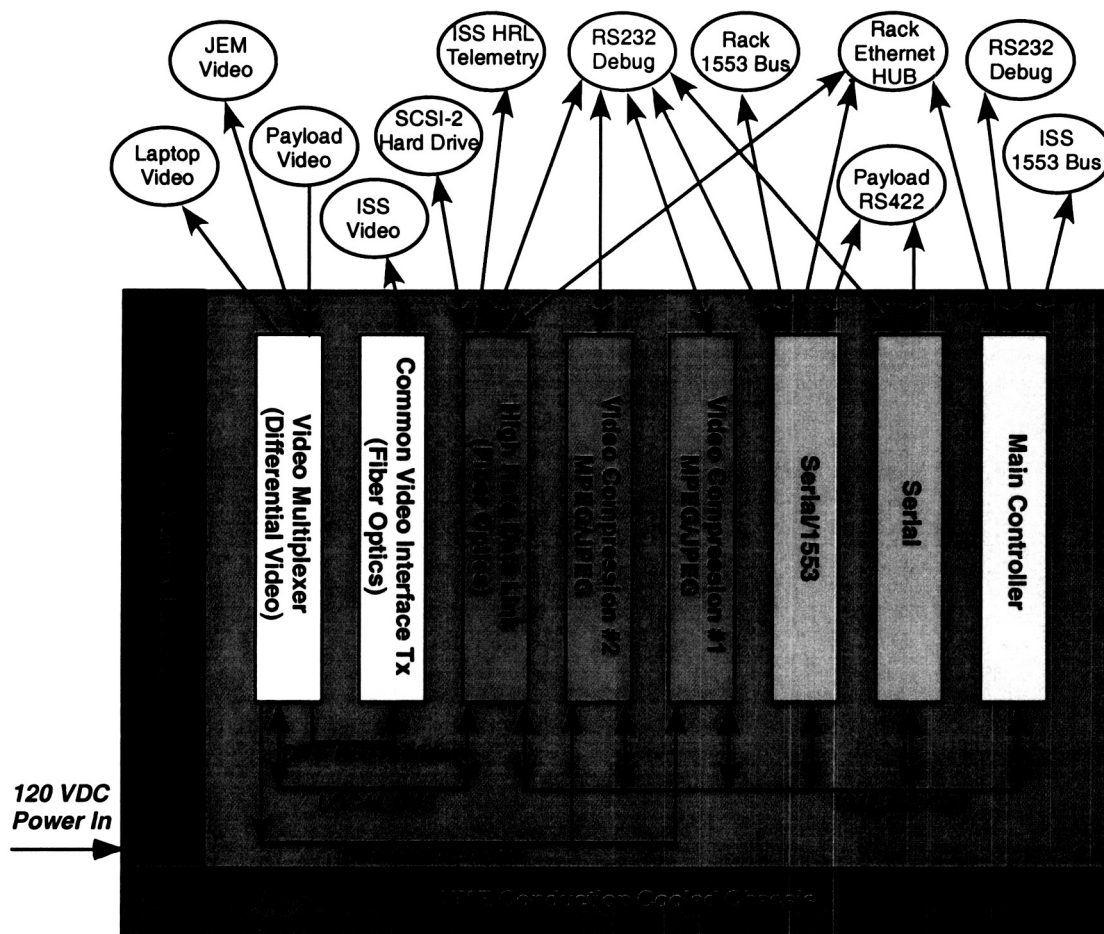


Figure 5: BRIC internal functional VME layout and input/output interface

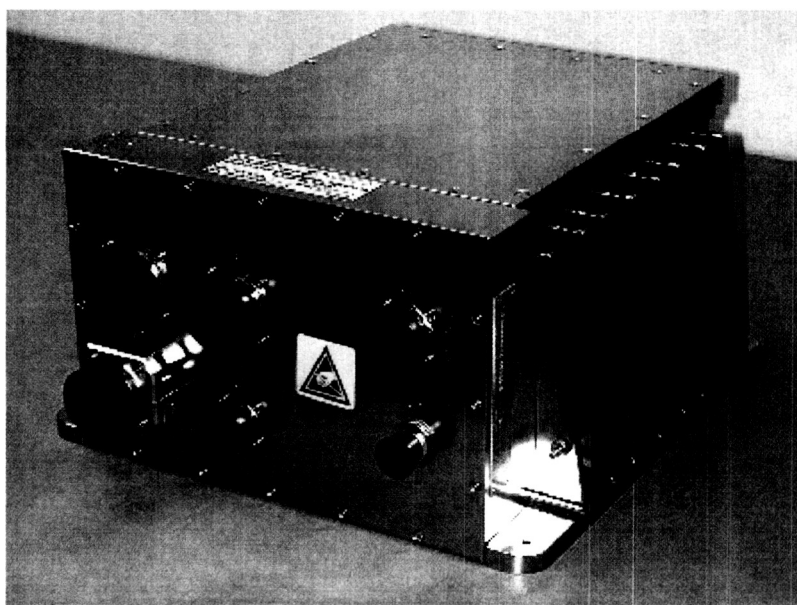


Figure 6: Fully populated BRIC in a conduction cooled chassis

The Biological Rack Interface Controller (BRIC) is an embedded computer used in the HHR rack, and represents the second generation of RIC. This configuration adds two VDCC (video digitization-compression card) to accommodate state-of-the-art MPEG-2 (Moving Pictures Expert Group) and JPEG (Joint Photographic Expert Group) video compression. Figure 5 depicts the internal BRIC input/output interface. Figure 6 is a fully populated BRIC with VME cards installed in a conduction cooled chassis enclosure.

The third generation RIC is the Centrifuge Rack Interface Controller (CRIC), and includes the Analog Devices "Digital Signal Processor to Video Digitization-Compression" card for audio compression. CRIC is an embedded computer used in the NASDA Centrifuge project.

RIC SOFTWARE ARCHITECTURE

The RIC CSCI consists of four TLCSCs (Top-Level Computer Software Components):

- Main Controller Card TLCSC
- Serial Card TLCSC
- Serial/1553 Card TLCSC
- High Rate Link Card TLCSC

CSCI architecture is based on utilization of the Wind River Systems VxWorks® operating system, a COTS (Commercial Off-The Shelf), real-time operating system widely used in space-based applications including the Mars Pathfinder and NASA Deep Space One. VME card communications are implemented using VxWorks® shared memory options, VxWorks/MP®, and direct memory access for data transfer across the VME bus.

The proven real-time capabilities of VxWorks® provide a cost-effective design approach for RIC software development, and flexibility for future hardware enhancements. Inherent capabilities of the VxWorks® operating system include message queues, real-time multitask scheduling, socket utilization, semaphores for interrupt service routines, and task synchronization modules, thus eliminating the need for customized low-level operating system design.

VxWorks® is hosted on the Main Controller card, Serial card, Serial/1553 card, and High Rate Link card. CVIT and Video Multiplexer cards do not contain RIC or kernel operating system software. The High Rate Link card video mux manager software controls the CVIT and Video Multiplexer cards through VME back-plane address and data lines. Figure 8 depicts the firmware/software layers on a typical RIC card.

The BRIC CSCI is a superset of RIC which adds the Video Digitization-Compression Card TLCSC for JPEG/MPEG video compression. Figure 7 depicts the BRIC CSCI decomposition and notates the common design elements across the different racks discussed herein.

The RIC CSCI serves as the bridge between ISS and the sub-rack payloads. It provides the command and control interface to the ISS C&DH system through the Payload Executive Processor MDM (Multiplexer/De-Multiplexer) via 1553 bus, collects science payload telemetry from RS422 or Ethernet interfaces, and downlinks via the selected telemetry path on low, medium, or high rate telemetry links. Additionally, RIC collects payload RS170 differential video and distributes to on-board laptop computers, and/or the video downlink system.

The BRIC system can digitize and compress payload video to MPEG-2 transport stream format or JPEG SPIFF (Still Picture Interchange File Format) format for real-time downlink (or for storage on the BEMU for later downlink). The RIC and BRIC CSCIs also interface to various sub-system components within HRF, EXPRESS, WOLF, and HHR racks, including PEHB, SSPCM, and the ARIS controller via rack-internal 1553 bus, providing data and power service to sub-rack payloads installed within the science research racks. Figure 9 illustrates the overall interface between ISS, rack avionics components, and sub-rack payloads.

Boeing utilizes a modified traditional waterfall model (illustrated in Figure 10) in developing the RIC software. This model assures the software product meet system software requirements early in the development and allowing for iterative software coding and/or prototyping for checkout of concepts used for requirements and design. HRF was the first completed rack, thus the CSCs and low-level drivers developed during this activity became the basis for software reuse in subsequent rack development efforts.

Both EXPRESS and its ARIS-derivative software were built based on HRF component reuse. Unique modules included an Avionics Air Assembly (AAA) command and control interface, ARIS controller interface, and minor modification to the RS422 and Ethernet manager CSCs to accommodate fewer payloads (from sixteen to ten). Approximately 97% of HRF software was reused in EXPRESS rack development. Likewise, WOLF rack software is 97% identical to EXPRESS. WOLF-unique modules include modifications to the thermal control system,

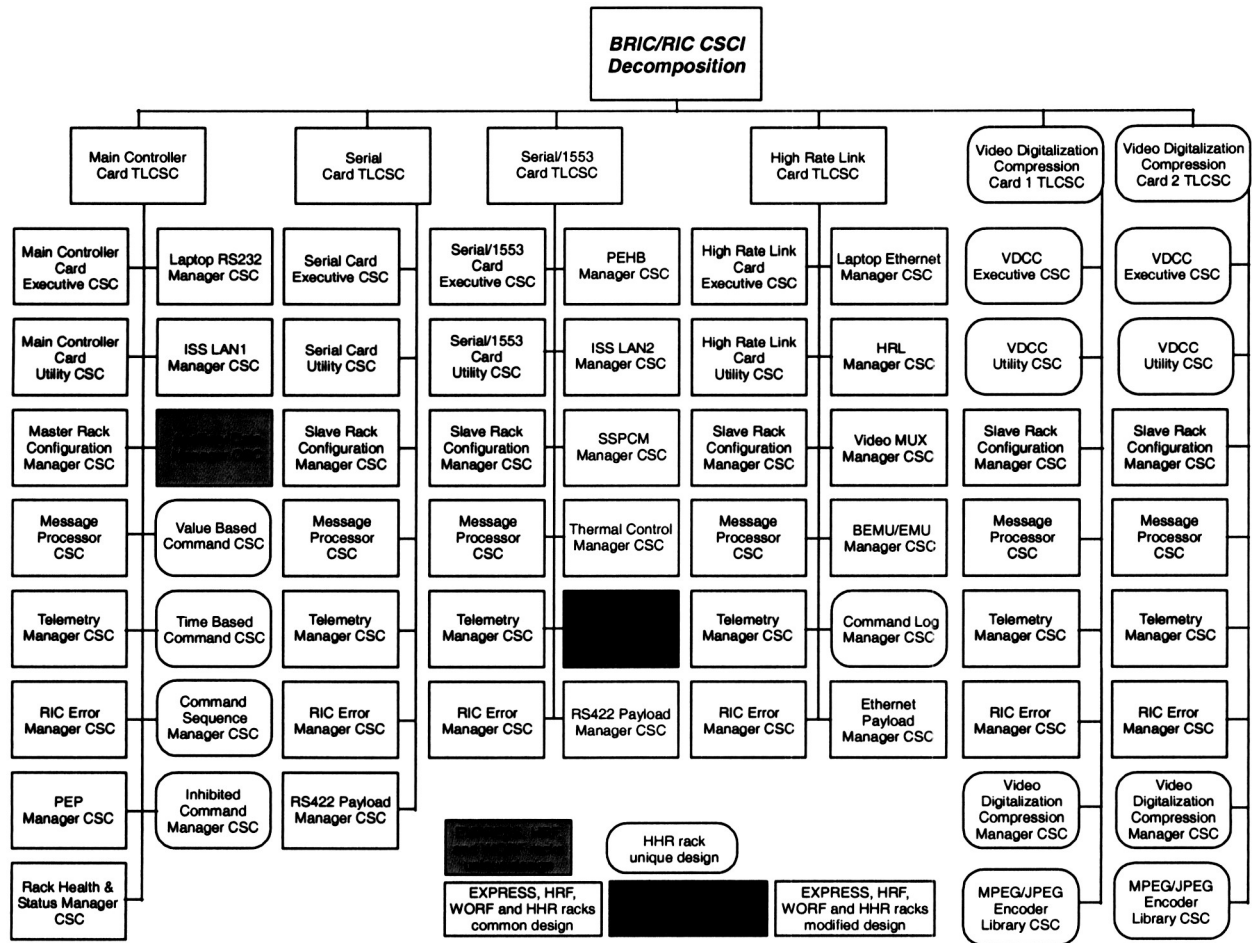


Figure 7: BRIC/RIC CSCI decomposition

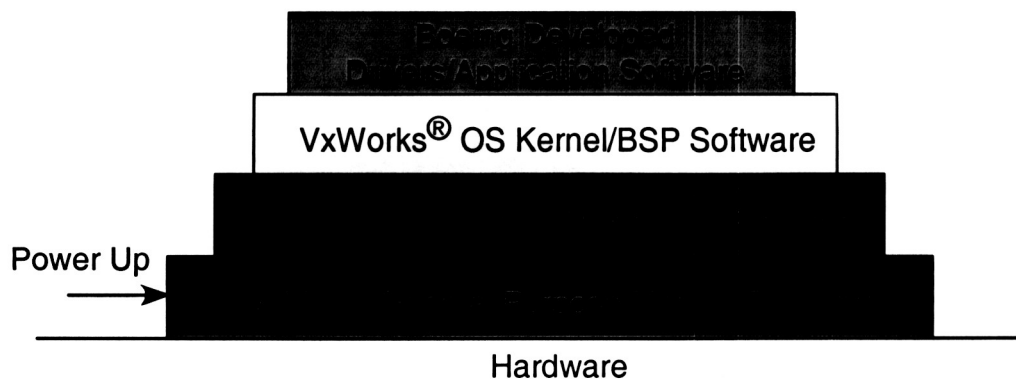


Figure 8: Firmware/Software layers on a typical RIC card

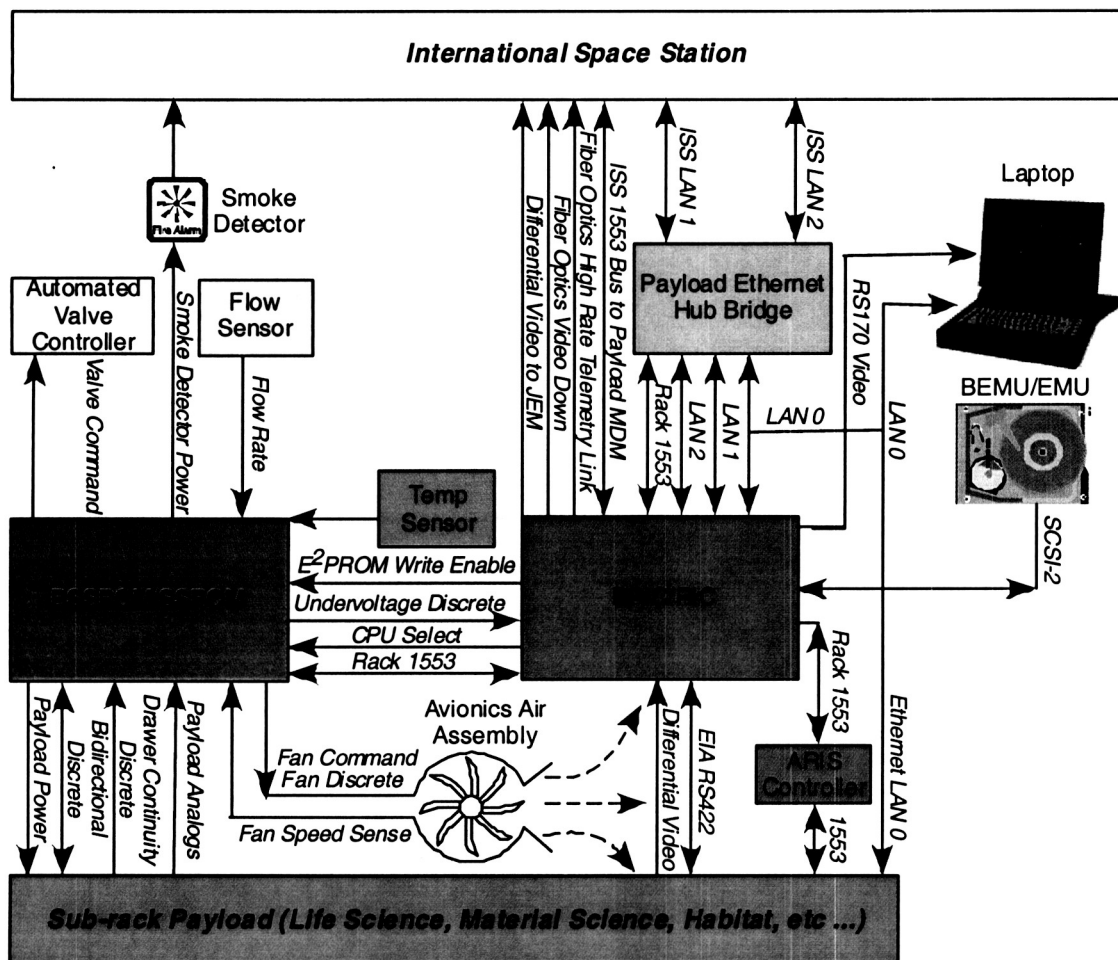


Figure 9: ISS, science research rack avionics components, and sub-rack payload interface diagram

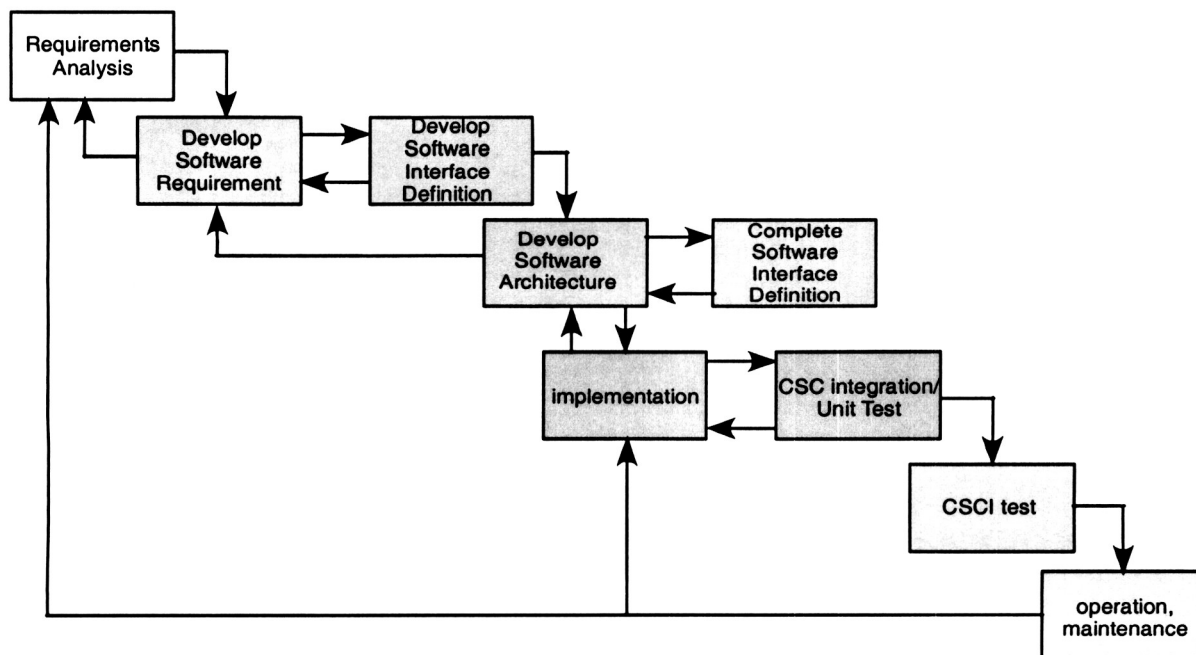


Figure 10: Modified Waterfall Model Software life cycle

support for five payload interfaces, and an interface to the BEMU.

In keeping with the theme of software reuse, BRIC was developed based on EXPRESS-RIC software and reuse of low level driver, message processor, telemetry processor, executive, and rack configuration manager modules to the maximum extent. The BRIC CSCI (a superset of RIC) implements enhanced end-user requirements for expanded capabilities, primarily in support of interface to the BSSPCM (Biological Research Project Solid State Power Controller Module for keep-alive, primary, and backup power requirements), BEMU for science telemetry storage requirements to accommodate Loss of Signal intervals, and video compression. Finally, CRIC software is approximately 98% identical to BRIC, with added capability for audio and video route-only functions on the video compression cards.

RIC SOFTWARE LESSONS LEARNED

HRLC ASIC Anomaly:

Boeing engineers identified anomalous behavior during the initial phases of HRF rack software integration on the HRLC (High Rate Link Card). The anomaly appeared during data transfer between RIC cards over the VME backplane. Investigation revealed that the HRLC ASIC (Application Specific Integrated Circuit), when operated on the VME bus in a slot other than the VME bus master, would lock up and begin writing to addresses over the bus.

The VxWorks/MP[®] option allows a group of cards to allocate and share common memory areas. Shared memory constructs may be accessed by any of the cards, but the VxWorks/MP[®] option ensures that only one card can access a construct at any given time. By default, the VxWorks/MP[®] option ensures shared memory resides on the Processor 0 card. It further guarantees memory integrity through the use of atomic operations or read-modify-write (RMW) cycles. An atomic operation locks out all other access until complete. RMW operations are performed when operating on any multiprocessor-safe constructs, such as shared message queues, shared semaphores, shared mutexes, or shared memory. For the Processor 0 card, the CPU performs atomic operations over the local bus. For other cards, the atomic operations occur over the VME backplane.

The ASIC anomaly appeared when a RMW cycle timed-out over the VME backplane. This occurred when the board was denied access to the VME bus

for long periods of time, typically during block transfers or repeated bus access by a VME board with higher bus priority. The ASIC entered a state in which it continually stepped through Processor 0 memory, writing to locations over the VME bus. This effectively locks the card and erases memory on Processor 0, resulting in a non-recoverable state.

Based on an analysis of technical robustness, cost, and schedule impact, Boeing determined that a software workaround to the ASIC anomaly was the preferred solution. The approach utilizes shared memory constructs when the system is in a known quiescent state, and a combination of VME bus interrupts to allow inter-processor communication. For example, if the card generates shared memory accesses that will not time out, the operation will succeed. When used in this manner, the only realistic time for shared memory operations is during software initialization. The following is a brief overview of the event sequences and the basic structure of a typical VME block transfer between the HRLC and Serial/1553 cards is illustrated in Figure 11.

Data Block Key

Violet	= allocated and in use by the system.
White	= marked as free.
Yellow	= allocated for use by the MCC card (marked as allocated, but not currently in use).
Green	= allocated for use by the SERC card.
Lt. Blue	= allocated for use by the S1553C card.
Red	= allocated for use by the HRLC card.
Rose	= allocated for use by the VDCC1 card.
Orange	= allocated for use by the VDCC2 card.

The sequence of events for a Block Transfer request from the HRLC to the S1553C is as follows:

- 1 HRLC retrieves the address of the managed data block that was allocated for use
- 2 HRLC retrieves the pointer to the allocated area from the block structure
- 3 HRLC transfers the notification message into the data block
- 4 HRLC transfers the message data into the data block
- 5 HRLC fills the send field in the event field for its entry on the destination card
- 6 HRLC generates a S1553C-unique VME interrupt, and awaits a SemTake for the S1553C to acknowledge receipt of the transfer

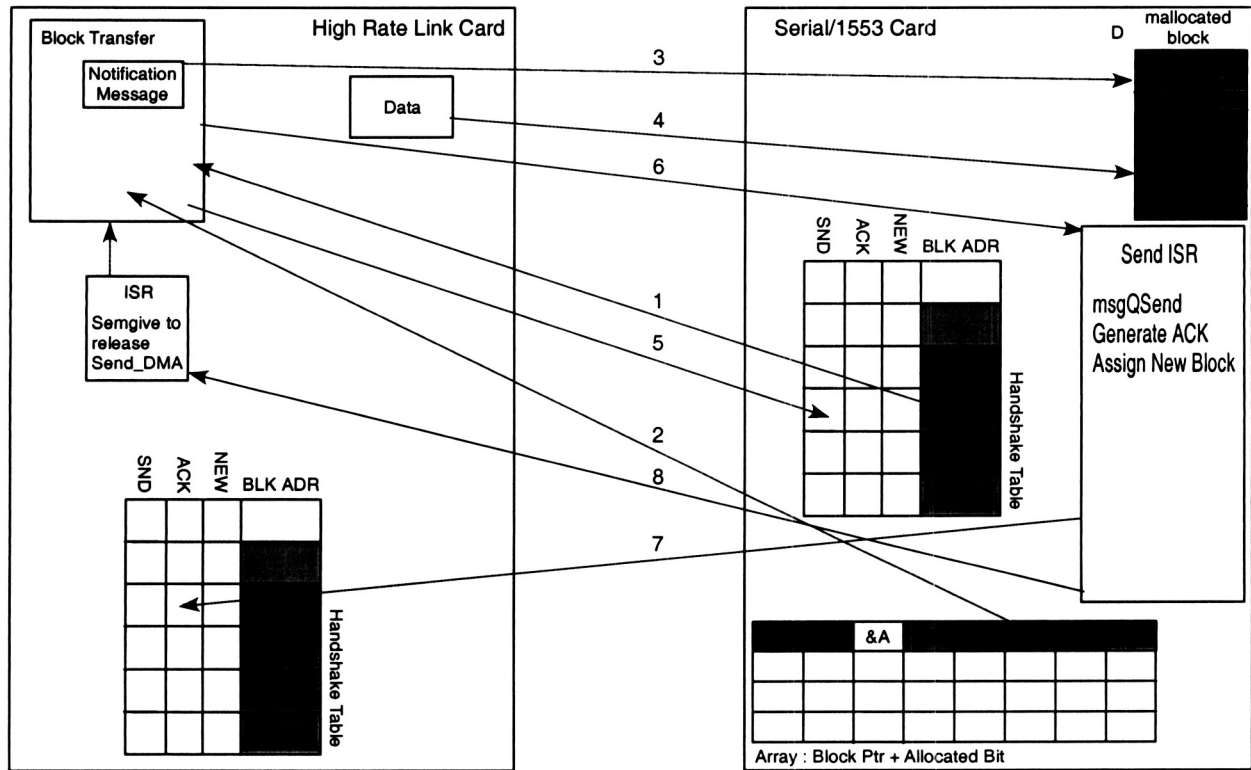


Figure 11: Typical DMA block transfer design between HRLC and S1553C cards

- 6.1 S1553C walks through the handshake table and finds the HRLC SND event set (it knows where the HRLC placed the NM and data from the S1553C handshake table)
- 6.2 S1553C generates a msgQSend to the proper message queue, which awakens the task for processing of the data
- 7 If successful, S1553C allocates a new block for HRLC and puts its address into the S1553C Handshake Table. If successful, the S1553C writes acknowledge OK to its location on the HRLC. Otherwise, it writes an acknowledge ERROR to its location on the HRLC.
- 8 S1553C generates an HRLC-unique VME interrupt to HRLC.
 - 8.1 HRLC searches through its handshake table and finds the S1553C ACK field set
 - 8.2 HRLC ISR releases the Send_DMA task with a SemGive
- 9 HRLC knows of the success or failure of the transmission and may act accordingly (This can be designed to have an interface similar to VxWorks®, where the caller could choose to WAIT_FOREVER, specify a number of retry attempts, or simply return ERROR if the transmission fails)

ISS LAN Restart:

The on-orbit ISS LAN restart anomaly that involved the Main Controller and S1553 Card is an excellent study for users of the AMD AM79C90 chip set with VxWorks®. Two issues are prominent with regard to the LAN restart problem:

- 1) During heavy system utilization, the Lance device experiences underflow conditions. Heavy system loading arises when there is significant activity on the local bus and prolific memory access. During a heavy-load scenario, the CPU, Lance, VIC/VAC VME controller chip, 68302 RS422 controller chip, and 1553 controller chip all demand system resources. The Lance hardware design relies upon a circular buffer contained in system memory. The circular buffer has a predefined format, including control, status, and data portions for each element in the ring. For transmission, the CPU writes outgoing data to the circular buffer, and the Lance reads out the data. An underflow condition can occur when the Lance FIFOs empty because data is transmitted onto the LAN faster than it can be received by the memory subsystem. The Lance detects the condition and generates an interrupt to the Lance device driver (Lance Restart), indicating a truncated packet. The Lance transmitter is

automatically disabled, pending intervention from the device driver, which is responsible for recovery.

- 2) During heavy LAN loading, external to the cards, the Lance device fails to transmit a packet sixteen successive times, and a retry error is indicated. The Lance device attempts to transmit a packet on the LAN, but is unsuccessful in negotiating ownership. After sixteen successive failures, the Lance automatically interrupts the device driver with a notice of the condition (Lance Restart). The Lance transmitter is automatically disabled pending intervention from the device driver, which again is responsible for condition recovery.

Unfortunately, the low-level software device drivers fail to recover properly from both conditions 1) and 2), resulting in a 'dead' Ethernet system until the card is rebooted. When the conditions above are manifest, Ethernet data ceases to appear on the LAN. To compound the problem, the application code does not receive any indication of a problem, and remains in the Ethernet transmit function call permanently.

To rectify this anomaly, Boeing procured the Lance device driver source code from Wind River Systems and embarked on a redesign effort to correct the application deficiencies, resulting in a driver capable of recovery from the conditions described above. Additionally, Boeing developed mechanisms to ensure continuous transmission of data when these events occurred. These measures guaranteed that data received by the device was actually transmitted, rather than relying on communication protocols to detect missing packets.

MCC Reboot:

During the ISS 6A mission, a Main Controller Card reboot problem occurred on EXPRESS Rack 2, attributable to a double-read of a message block pointer on the 1553 chip set (within the remote terminal interrupt service routine). Analysis determined that the second read of a message block pointer (when the pointer wraps around) causes an arithmetic overflow condition. This condition ultimately caused the interrupt service routine to jump to an illegal address, attempting to fetch a 1553 data word. This condition raised an exception in the interrupt service routine and caused the main controller card to reboot. A software solution was developed to preclude the occurrence of the double-read, and the common 1553 driver code was integrated into HRF, WORF, and HHR racks.

Interrupt Service Routine Exceptions:

Developing a software solution to handle exceptions raised within an interrupt service routine proved to be one of the most challenging problems encountered during HHR software integration. The occurrence of a VME bus error inside a normal task causes it to be suspended but easily resumed with proper detection and recovery code; however, a bus error within an interrupt service routine was typically rectified only by reboot. Most ISS payloads are continuously powered, performing scientific experiments over long duration cycles. Obviously, card reboots for devices controlling these experiments spell major impact to the affected rack payload, and could ultimately threaten mission success. Boeing again developed a novel software solution to handle the card reboot issue in the form of a VME bus error handler that 'intercepts' the interrupt service routine. By clearing the bus error bit on the VME controller and allowing continuous operation of the VME bus, code execution inside the interrupt service routine can resume. Again, the commonality of DMA Utility code across racks necessitated the installation of correction code in HRF, EXPRESS, and WOLF.

On-orbit Software Upload/Upgrade:

On-orbit upgrade to RIC software is currently accomplished by uploading via EXPRESS laptop computers configured with an RS232 port, PROCOMM communication software, and a Quatech four-serial-port PCMCIA (Personal Computer Memory Card Interface Architecture) card. This approach loads S-record images into the RIC flash memory devices. Due to low RS232 bandwidth, software upload processes are cumbersome at best, consuming about one hour of (costly) on-orbit crew time.

In addition to the time-consuming, crew-intensive nature of the process, the potential to introduce error during serial port configuration of the EXPRESS laptop and RIC is high. A new method of uploading RIC software is under consideration (which does not involve crew intervention). The improved approach implements a 1553 flash memory software loader, together with a process whereby RIC software is uploaded to the EMU/BEMU via PEP MDM file transfer utilities and transferred to individual cards over VME bus into flash memory. This new concept eliminates crew intervention to mate and de-mate cables at the laptop interface. From ground facilities at MSFC's Huntsville Operations Support Center, the Payload Rack Officer can manage and perform all rack software upload operations.

SUMMARY

The Payload rack configurations described herein exhibit high degrees of commonality in both hardware and software. Reusable common software has reduced development flow time, improved reliability, and significantly lowered software life cycle and maintenance cost. Additionally, software lessons-learned identified in the design/verification stages of initial product development efforts can be shared and applied to later designs to improve software reliability and reduce cost. System performance has been optimized over several hardware interfaces, serial and Ethernet communications modules have received significant performance and robustness improvements. By applying significant effort in the initial software design phases, a modular product structure has evolved which supports differing applications and platforms with predictable, reliable, high-quality results. As we continue our support to the ISS mission of long duration space and science exploration, our efforts are directed toward increasing the maintainability, reliability, and low-cost performance of on-orbit science research projects through advanced software and hardware technology.

ACKNOWLEDGMENTS

I would like to thank Mr. Kevin Pierce and Mr. Keith Stanley who spent numerous of hours to review this paper. This paper would not be completed without their valuable editorial feedback and comment.

REFERENCES

- [1] EXPRESS Racks 1 and 2 Fact Sheet Number FS-2001-02-34-MSFC
- [2] Human Research Facility Rack Fact Sheet Number FS-2001-02-43-MSFC