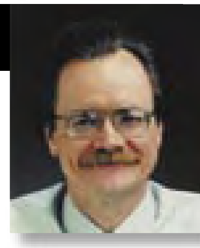

The quickest way to compliant and interoperable Intelligent Platform Management for AdvancedMC carriers and modules

By Mark Overgaard

The Advanced Mezzanine Card (AdvancedMC or AMC) architecture further strengthens the burgeoning AdvancedTCA market with a hot swappable mezzanine form factor designed from scratch to address the needs of telecom platforms based on AdvancedTCA.

In this article Mark focuses on the management aspects of the AdvancedMC base specification, AMC.0, which is nearing completion within PICMG. AMC.0 extends the definition of an AdvancedTCA IPM Controller (IPMC) with additional responsibilities to function as an AMC carrier IPMC, representing its installed AMC modules to the shelf manager. AMC.0 also defines a simplified Module Management Controller (MMC) for the AMC modules themselves.

In order to simplify and reduce the cost of MMCs, AMC carrier IPMCs have considerably more responsibility than ordinary AdvancedTCA IPM controllers. The AMC.0 specification devotes more than 60 pages to defining the requirements on these management subsystems. Implementing these requirements requires a thorough understanding of both the initial AdvancedTCA specification, having more than 160 pages about management, and the underlying Intelligent Platform Management Interface (IPMI) specification, having 420 additional pages of requirements.

Fortunately, there's an alternative to developing all this technology from scratch. Off-the-shelf management subsystems, such as the IPM Sentry components, can simplify and speed the development of both the carrier IPMC and the MMC. Using an off-the-shelf design for these components allows developers to focus on their unique added values rather than developing and validating the management subsystems on their own.

Please see "Platform management for PICMG architectures" in the *CompactPCI Systems PICMG Resource Guide* for an overview of IPMI-based AdvancedTCA board management. Also, please see "Advanced Mezzanine Card fulfills the promise of AdvancedTCA," *CompactPCI Systems,* October 2003 for an introduction to the goals and features of the AdvancedMC form factor.

AdvancedMC management: Adding a layer to AdvancedTCA shelf management

Figure 1 shows how AMC management relates to the existing AdvancedTCA shelf management framework. From the point of view of the shelf manager, an AMC carrier is just another AdvancedTCA board, represented by an IPM Controller and accessed over the dual redundant IPMB-0. The AMC.0 specification is focused on AdvancedTCA AMC carriers, but other possibilities exist; the specification aims to facilitate such alternatives without attempting to address them directly. One such alternative is an AMC backplane that directly accepts AMCs without an intervening carrier. This alternative may be addressed by its own specification(s) in the future.

The AMCs on such a carrier are considered as managed Field Replaceable Units (FRUs) and seamlessly fit into the AdvancedTCA management framework for hot swappable FRUs.

A carrier IPMC uses IPMB-L ("L" for local) to communicate with the MMCs on its AMC modules. Like IPMB-0, IPMB-L can be implemented either as a single bus or as a set of radial buses, one for each AMC bay. If IPMB-L is a single logical bus, there are isolators at each AMC site to ensure that a malfunctioning MMC cannot disrupt the operation of the entire IPMB-L.

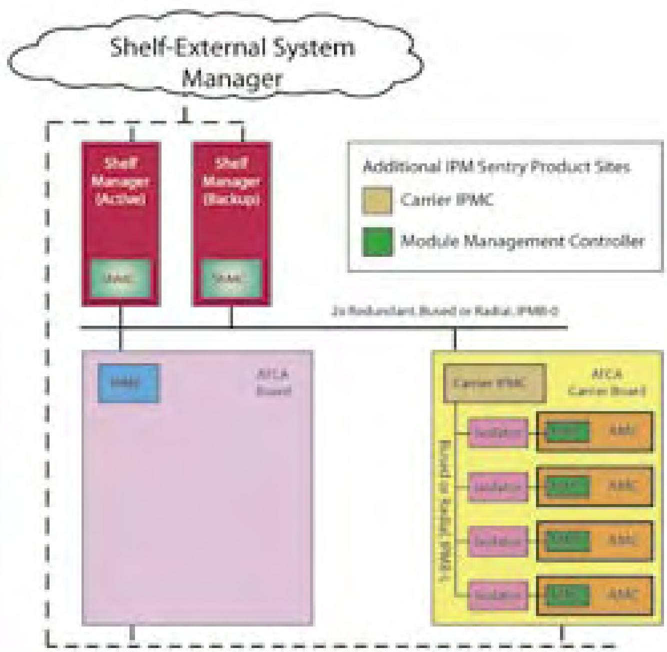

Figure 1

The key goals for the AMC management infrastructure include:

- Fit smoothly into established PICMG 3.0 conventions. AMC management aims to be an extension of AdvancedTCA management rather than a new and separate architecture.
- Avoid impacting PICMG 3.0 R1.0 shelf managers with AMC.0. By design, AMC carriers can be installed in existing AdvancedTCA shelves; no AMC-specific knowledge is required of the shelf manager. AMC-awareness is an advantage for AdvancedTCA shelf managers, but not a requirement.
- Reduce requirements on MMCs to limit its cost and footprint. For instance, AMC.0 defines a simplified hot swap and power management architecture for AMC modules.
- Require carrier IPMC to represent MMCs as full-fledged AdvancedTCA FRUs to the shelf manager. For example, carrier IPMCs negotiate power requirements with the shelf manager on behalf of AMC modules. Each MMC defines a single static power requirement; the carrier IPMC maps this to the dynamic multilevel AdvancedTCA power negotiation architecture.
- Preserve the IPMI foundation in MMCs. For instance, the full IPMI sensor infrastructure is optionally available for MMC sensors.

The remainder of this article shows how to quickly realize carrier IPMCs and MMCs with off-the-shelf components, using the IPM Sentry BMR-AVR-AMC reference design as an example.

AMC management hardware design

On the hardware side, the IPM Sentry BMR-AVR-AMC reference design addresses both carrier IPMC and MMC functions. The design is delivered as a pair of schematic and bill of material, ready for immediate integration into the design of an AdvancedTCA AMC carrier or AMC module. The carrier IPMC is based on the IPM Sentry BMR-AVR-AdvancedTCA second generation IPM Controller design, which incorporates a pair of highly integrated Atmel AVR microcontrollers. The carrier IPMC adds a third AVR microcontroller (the carrier AVR) to handle the additional responsibilities of the carrier role. A single AVR microcontroller serves as the MMC in this design.

Figure 2 shows the BMR-AVR-AMC design as implemented on the BMR-AVR-AMC bench top prototype board. The IPM Controller is not detailed in the diagram, which focuses on AMC-specific functionality.

The carrier controller block includes the carrier AVR and one AMC site (typically one of several AMC sites on a real carrier). The key active components in this block are:

- The carrier AVR, which integrates Flash memory, EEPROM, and numerous peripherals, such as Analog-Digital Converters (ADCs) and GPIOs. The carrier AVR is an ATmega64 or ATmega128. The carrier AVR handles the additional responsibilities of a carrier IPMC compared to an ordinary AdvancedTCA IPM Controller. It uses IPMB-L to communicate with the AMCs and a master-only I2C bus to communicate with the various control and monitoring functions at each AMC site. Serial Peripheral Interface (SPI) links the carrier AVR to the rest of the IPM Controller.
- Supplementary SRAM, which is especially important for storing the Sensor Data Records (SDRs) of the AMCs. The carrier AVR maps these SDRs so that the Shelf Manager sees them as owned by the carrier IPMC.

Each AMC site has the following key active components:

- An IPMB buffer (Linear Technology LTC4300A-1) to isolate IPMB-L from each AMC site. A given site is enabled for IPMB-L communication only if an AMC is present and not judged to be disrupting IPMB-L.
- Payload and management power controllers to ensure that specification- and/or design-defined limits on power to each AMC site are respected.
- An ADC to monitor the voltage and current of the payload power, plus the voltage of the management power, that are supplied to the AMC on each site.
- An I2C GPIO array to manage the power controllers and IPMB-L buffer.

Finally in Figure 2, there is the MMC, with the following key components:

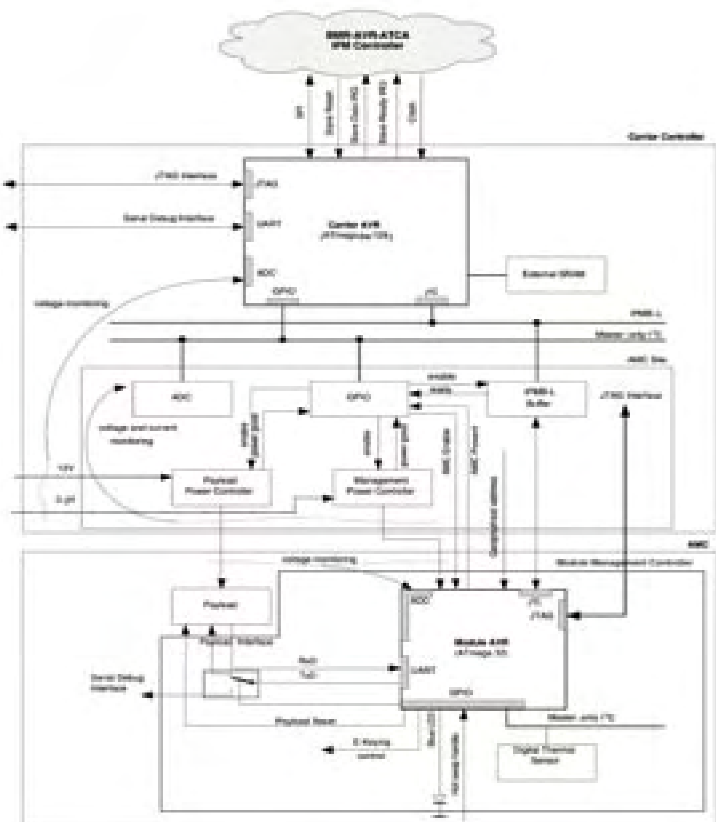

Figure 2

- The module AVR, an ATmega32 that integrates Flash and numerous peripherals. GPIOs on the module AVR implement many functions, including reading the geographic address of the AMC site and implementing the hot swap interface with blue hot swap LED and handle switch. Optionally, a latch can be added to preserve the key control outputs (the payload reset signal, for instance) across MMC resets.
- A thermal sensor. This is assumed in Figure 2 to be digital and accessed over I2C, but the sensor could be an analog one, monitored via an AVR ADC.
- AVR ADCs also monitor the payload and management power feeds. The figure assumes that the needed measurement accuracy can be achieved with the built-in voltage reference, but an external voltage reference is an alternative, as well.
- A simple switch that allows the AVR to control the sharing of a single UART between the payload interface and an output-only serial debug interface. In some designs, only one of these interfaces will be implemented, and the switch will be dropped.

IPM Controller firmware

The BMR-AVR-AMC firmware augments the BMR-AVR-AdvancedTCA firmware to support a compliant Carrier IPMC on the three Carrier-based AVRs, as well as compliant MMCs for AMC modules. In addition to the obvious advantages of using off-the-shelf firmware that has already been confirmed as compliant and interoperable, the BMR-AVR firmware speeds the development process with a highly streamlined process of configuration for a custom hardware environment.

Except for one source file and the two data files mentioned later in this article, the BMR-AVR firmware can typically be used as is on a custom AMC or carrier, providing immediate savings in development time. That one source file (config.h) provides configuration guidance for the rest of the firmware on either the carrier or AMC. For instance, one line of that file for a carrier defines how many AMC sites are implemented on the carrier. Similarly, one-line changes in that file for an AMC can configure firmware support for analog or digital temperature sensors in or out of the MMC firmware image.

In the area of sensors, the BMR-AVR firmware architecture (cleverly leveraging the facilities of IPMI as well) further reduces development effort by providing simple and quick configuration mechanisms for board-specific adaptations. Consider, for instance, a carrier or AMC that needs to provide Shelf Manager visibility to some board-specific voltage. The AVR microcontrollers include built-in ADCs; one of these can be allocated to monitor this voltage. One line in a *config.h* file accomplishes this allocation.

In addition, it is necessary to describe the IPMI properties of this sensor via a Sensor Data Record (SDR). The SDR can define, for instance, a critical threshold voltage for this sensor and configure it to trigger an event to the Shelf Manager when the voltage crosses that threshold going down. The SDR also indicates how the raw binary readings from the ADC are converted to human-readable voltage units.

The SDRs for all the sensors implemented on a particular carrier or AMC are collected in an *sdr-data.bin* file that is loaded on the board. An SDR compiler supplied with the IPM Sentry firmware produces this file from simple textual representations of those SDRs. The single line in the *config.h* file mentioned earlier that assigns one of the built-in ADCs to monitor a voltage also identifies the SDR that defines the properties of this voltage sensor.

The final part of the adaptation process is to create appropriate FRU information (*fru-info.bin)* for the custom Carrier or AMC, including both IPMI-specified data structures, such as product name and AdvancedMC extension data structures, such as descriptors of the point-to-point links implemented on the Carrier or by an AMC (for use in E-Keying). A supplied FRU compiler produces *fru-info.bin* from simple textual representations of this data.

With this architecture, a developer can do sophisticated adaptations to board-specific features without creating *any* C language code other than a few simple lines in the *config.h* file. Of course, full source code of the firmware is available for developers who wish to do more extensive customizations.

IPM Sentry bench top hardware

AdvancedTCA is still new, and AdvancedMC is even newer (with the latter specification not yet complete). As a result, most AMC developers are new to the specification and especially to the management aspects, even if they are previously familiar with IPMI or AdvancedTCA management. Bench top implementations of the IPM Sentry Shelf Manager and BMR-AVR-AMC carrier IPMC and MMC provide immediate hands-on access to a compliant management subsystem.

Figure 3 shows a bench top BMR-AVR-AdvancedTCA IPM Controller board, cabled to a BMR-AVR-AMC bench top board. Consistent with Figure 2, the latter board adds a Carrier Controller and two simulated AMC sites, each with carrier-based control facilities and an MMC. This pair of boards represents an AdvancedTCA AMC carrier with two AMC sites. Using a flat ribbon cable to implement IPMB-0, this bench top *AMC Carrier* can be linked to a second BMR-AVR-AdvancedTCA IPM Controller bench top board and a bench top IPM Sentry Shelf Manager. This three-node management subsystem enables developers to explore the behavior of AdvancedTCA and AdvancedMC management facilities, before *any* custom hardware is available.

Here are some of the development activities that are facilitated by a bench top shelf management configuration including the *AMC Carrier* like that in Figure 3, plus a bench top Shelf Manager:

- General familiarization with AdvancedTCA and AdvancedMC platform management using live hardware and software rather than having to rely solely on 220+ pages of dense specification language.

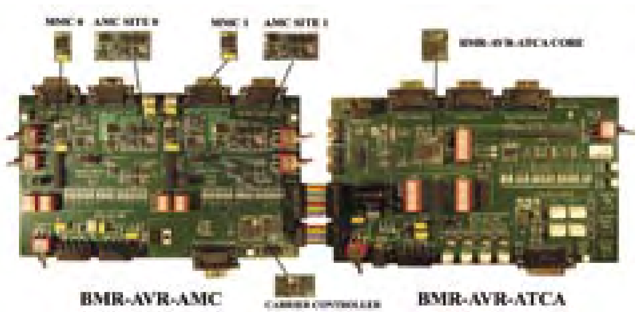

Figure 3

- Using on-board headers and switches to confirm understanding of the operation of hardware and software while finalizing design plans.
- Experimenting with hardware and software options in the reference design, while finalizing decisions about mapping the reference design into a custom board.
- Using the bench top AMC carrier and pair of AMC sites as a *known good* reference, while bringing up a custom board.
- Using the bench top shelf manager during bring up of the custom board, possibly even cabled into a development shelf.

Conclusion

The quickest way to a compliant and interoperable Intelligent Platform Management for AdvancedMC carriers and modules is to use off-the-shelf components. The IPM Sentry family of components is developed and backed by experts in PICMG management and include schematics, source code, bench top hardware, and comprehensive documentation. Using these components can remove the platform management aspects of an AMC-related project entirely from the critical path, allowing developers to focus their energy on the unique value-added functionality of their boards.

Mark Overgaard *founded Pigeon Point Systems (PPS) in 1997 to focus on products and services supporting the adoption of open modular platforms to replace proprietary architectures, with an initial focus on the telecommunications market and CompactPCI. He is a leader in the technical subcommittees of PICMG (including the management aspects of AdvancedTCA and the corresponding CompactTCA specification, now in development). Previously Mark was vice president of engineering at Lynx Real-Time Systems (a UNIX-compatible RTOS supplier), and TeleSoft. He earned an MS in Computer Science from UC San Diego and a BS in Physics from Geneva College.*

For further information, contact Mark at:
Pigeon Point Systems
P.O. Box 66989
Scotts Valley, CA 95067-6989
Tel: 831-438-1565
E-mail: mark@pigeonpoint.com