The quickest way to a compliant and interoperable Intelligent Platform Management Controller for AdvancedTCA

By Mark Overgaard

Rapidly building interest in the AdvancedTCA architecture has spurred numerous developers of AdvancedTCA boards to investigate meeting the specification's management requirements. These requirements also interest developers of other management-controller-equipped (*intelligent*) Field Replaceable Units (FRUs), such as fan trays, power entry modules, and the like.

Adopted on the last day of 2002, the original PICMG 3.0 specification includes more than 130 pages of management-related material. The first revision (just adopted in January 2004) increases the shelf management material to more than 160 pages. The AdvancedTCA management architecture leverages the Intelligent Platform Management Interface (IPMI) specification that accounts for another 420-plus pages of requirements. Fully complying with these specifications creates a serious engineering project, even if a company's existing subsystem supports the IPMI specification.

Fortunately, there's an alternative to developing all this technology from scratch. Off-the-shelf management subsystems, such as the IPM Sentry components, can dramatically simplify and speed Intelligent Platform Management (IPM) Controller development. Using an off-the-shelf IPM Controller design allows developers to focus on their unique value adds rather than developing and validating the management subsystems on their own.

Please see "Platform management for PICMG architectures," in the 2004 PICMG Resource Guide page 138 for an overview of IPMI-based platform management for AdvancedTCA, CompactPCI, and the upcoming CompactTCA architectures.

AdvancedTCA board and IPM Controller interfaces

Figure 1 shows an AdvancedTCA board's key interfaces, including the interfaces to the logical IPM Controller subsystem. The inset block in the upper right corner shows the approximate scale size of an IPM Sentry BMR-AVR-based IPM Controller (one off-the-shelf solution for an AdvancedTCA management subsystem) as it could be implemented on an AdvancedTCA board.

An AdvancedTCA IPM Controller manages the local aspects of one or more FRUs and represents those FRUs to the Shelf Manager. The IPM Controller responsibility areas for AdvancedTCA board FRUs (see Figure 1) include:

- Basic IPMI management: Supplies inventory data (serial number and so on), sensor visibility, and event notifications via in-shelf IPMB-0.
- Payload interface: Creates a communication channel between the IPM Controller and the payload (or main) part of the board.
- Power and cooling: Control and monitor power for the board's payload and monitor at least one temperature sensor.
- Backplane interconnects: Following Shelf Manager *E-Key* guidance, enable and disable point-to-point interfaces between the board and the backplane and coordinate use by the board's payload of bused backplane resources (such as synchronization clock buses) with other boards' use of the resources.

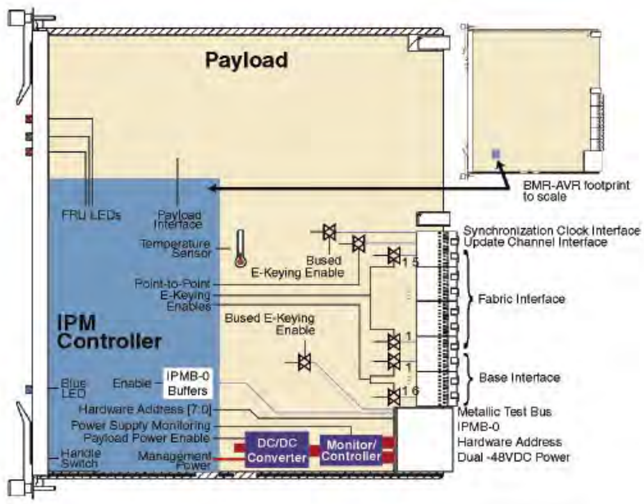

Figure 1

The quickest way to IPM Controller hardware design: Off-the-shelf IPM Sentry

On the hardware side, an IPM Sentry IPM Controller reference design arrives initially as a schematic and bill of material, ready for immediate integration into the design of an AdvancedTCA board or other intelligent FRU. The IPM Sentry BMR-AVR second-generation IPM Controller design incorporates a pair of highly integrated Atmel AVR microcontrollers. The IPM Controller's core can fit within an 18mm by 23mm area, so developers need invest very little board real estate in the management subsystem. Figure 2 shows the BMR-AVR design as implemented on Pigeon Point Systems' bench top prototype board. The active components in the IPM Controller are:

- Master and slave AVR microcontrollers, each of which integrates Flash memory, SRAM, EEPROM, and numerous

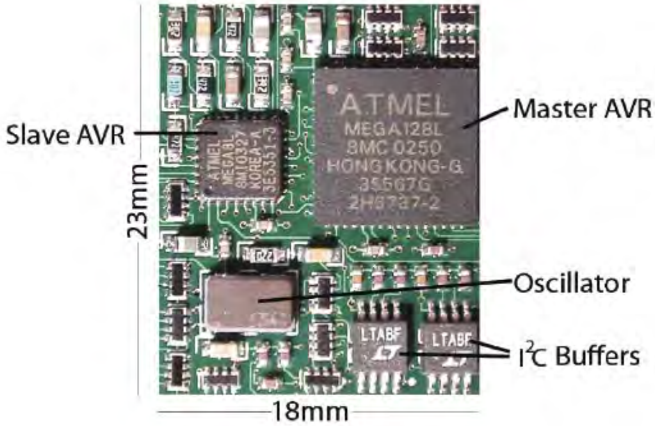

Figure 2

peripherals, such as Analog-Digital Converters (ADCs) and GPIOs. The master AVR is an ATmega64 or ATmega128 (depending on whether the relevant Board Manager software needs 64 or 128 Kbytes of Flash). The slave AVR is an ATmega8 with 8 Kbytes of Flash.

- Dual IPMB buffers (Linear Technology LTC4300A-1s) to isolate the IPMC from the backplane's dual redundant IPMB-0. IPMB-0 is I^2C-based and provides the primary shelf-internal communication path between the shelf's IPM Controllers and the overall Shelf Manager.
- An oscillator to supply a precise common clock for both AVRs. If a particular design doesn't require a precise operational frequency, on-chip oscillators can be used instead.

Figure 3 shows how the two AVRs cooperate to enable some of the key functionality of an IPM Controller, including the following:

- UART-based Payload and Serial Debug interfaces supported by the master AVR.
- Control and monitoring of the I^2C buffers between the IPM Controller and the backplane IPMB-0.
- Optional use of a latch buffer to ensure that if the IPM Controller resets for any reason (say, due to a watchdog expiration restart), it does not affect key signals. Controls that can be protected in this way include the blue hot swap LED, the E-Keying controls for Base and Fabric Interfaces imple-

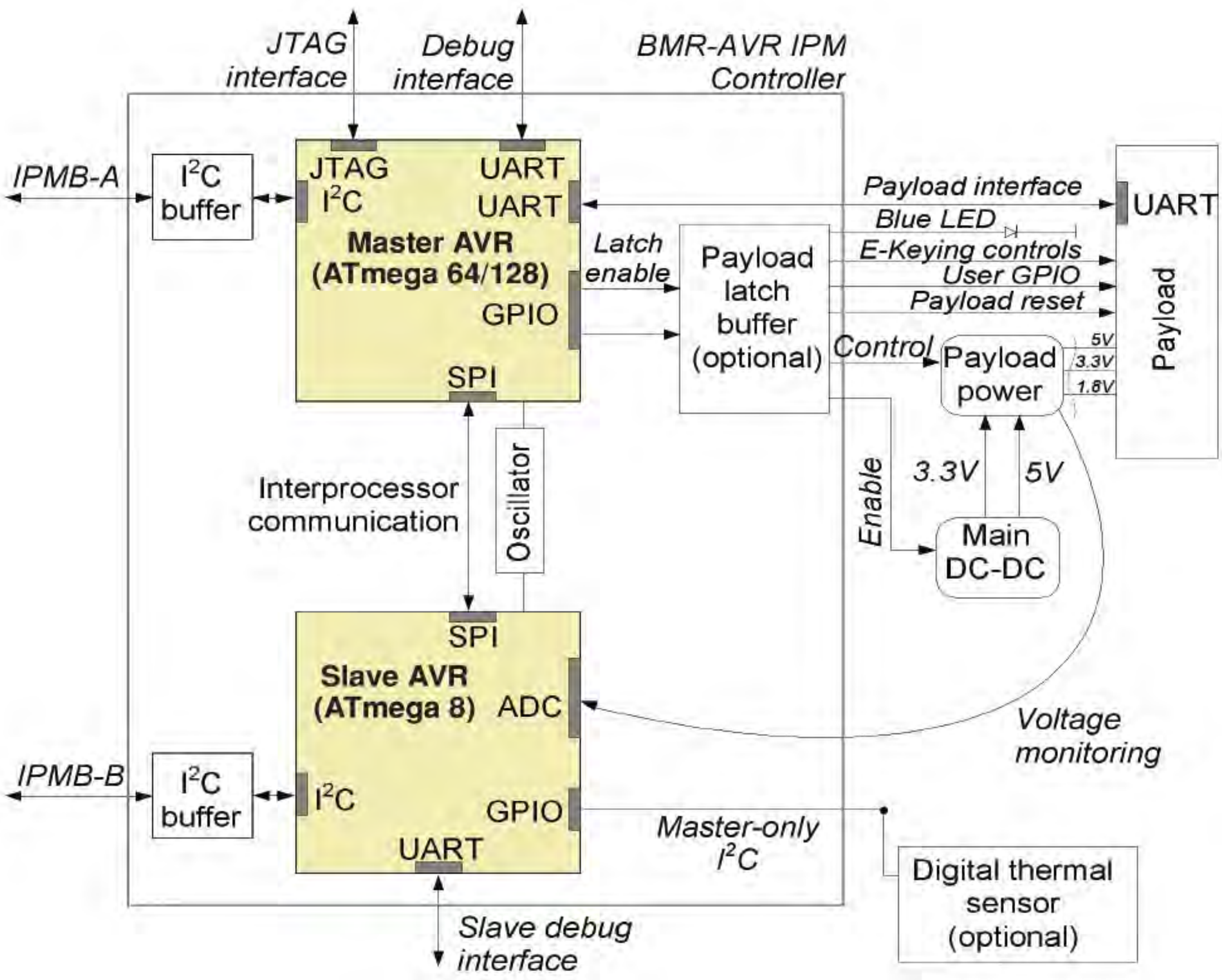

Figure 3

mented by the board, and controls for various payload power aspects.
- Payload voltage monitoring performed by ADCs on the slave AVR.
- Interprocessor communication between the two AVRs via the AVRs' three-wire Serial Peripheral Interface (SPI) synchronous serial ports plus cross-connected interrupt signals.

The quickest way to IPM Controller software: Off-the-shelf IPM Sentry

The IPM Sentry Board Manager software, provided in source form, runs on the BMR-AVR hardware to create a compliant, interoperability-tested IPM Controller. Figure 4 shows the main layers of the Board Manager:

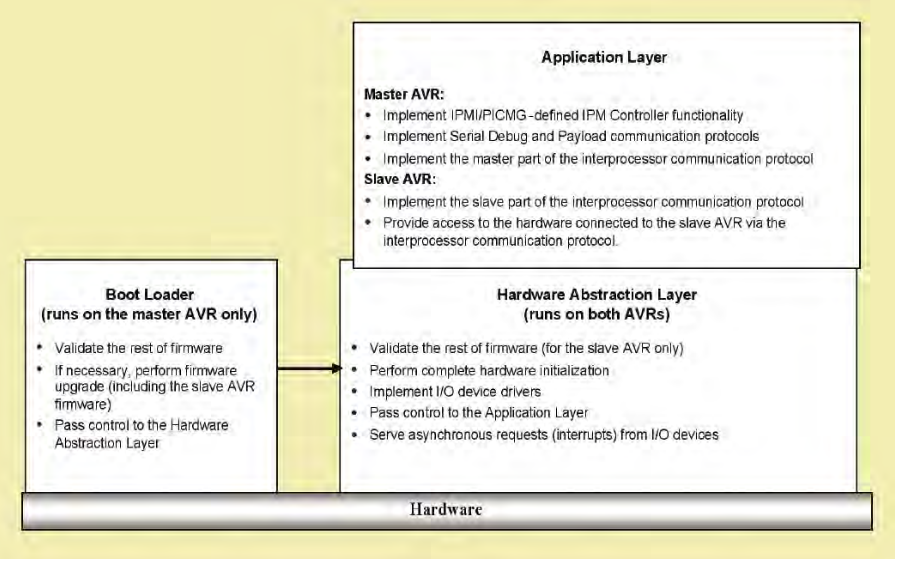

Figure 4

- Boot Loader: Determines whether the firmware needs to be upgraded and performs the upgrade if necessary.
- Hardware Abstraction Layer: Provides application-independent high-level interfaces to the low-level device drivers for the various controllers in the AVR hardware.
- Application Layer: Implements the IPMI and PICMG defined functionality of an IPM Controller and the IPM Sentry extensions, such as the serial interface protocol that the Payload and Serial Debug interfaces use.

Designers can typically use the IPM Sentry Board Manager software as is on a custom board to immediately save development time. However, IPM Sentry Board Manager software use does require board-specific updates to one source file and two data files. The source file (config.h) provides configuration guidance for the rest of the Board Manager. For instance, one line of that file enables or disables the fan management subsystem (typically not needed on a generic 8U AdvancedTCA board) of the Board Manager. If that subsystem is disabled, all software related to it is entirely omitted when the Board Manager is built for loading on the board.

In the area of sensors, the Board Manager architecture (cleverly leveraging IPMI facilities as well) further reduces development effort with its simple and quick configuration mechanisms for board-specific adaptations. Consider a board that needs shelf manager visibility to some board-specific voltage. The AVR microcontrollers include built-in ADCs, and users can allocate one of these to monitor this voltage. One line in the config.h file accomplishes this allocation.

In addition, describing the IPMI properties of this sensor requires 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 convert to voltage units that can be read by users.

The SDRs for all of a particular board's sensors are collected in that board's sdr.bin file. An SDR compiler supplied with the IPM Sentry Board Manager produces this file from simple textual representations of those SDRs. The single line in the config.h file mentioned above that assigns one of the built-in ADCs to monitor a voltage also identifies the SDR that defines this voltage sensor's properties.

Creating appropriate FRU Information (fru-info.bin) for the custom board completes the adaptation process. This information includes both IPMI-specified data structures, such as product name, and AdvancedTCA extension data structures, such as descriptors of the point-to-point links to the backplane implemented by the board (for use in E-Keying). A supplied FRU Compiler produces fru-info.bin from simple textual data representations.

With this architecture, a developer can make sophisticated adaptations to board-specific needs without creating any C language code other than a few simple lines in the config.h file.

Immediate hands-on access: IPM Sentry bench top hardware

Because AdvancedTCA is new, most AdvancedTCA board developers are new to the specification and especially to the management aspects, even if they are previously familiar with IPMI. Bench top implementations of the IPM Sentry Shelf Manager and BMR-AVR IPM Controller make possible immediate hands-on access to a compliant management subsystem. Figure 5 shows the IPM Sentry ShMM-300 Shelf Management Mezzanine installed on a bench top carrier board to create a bench top Shelf Manager. Figure 5 also shows one of the two bench top IPM Controller boards that are shipped to new IPM Sentry board management customers. With a flat ribbon cable implementing IPMB-0, this collection of boards enables rapid ramp-up on AdvancedTCA management's hardware and software aspects in advance of custom hardware availability.

A bench top shelf management configuration such as that in Figure 5 facilitates development activities, including:

- Gaining general familiarity with AdvancedTCA platform management using live hardware and software rather than having to rely solely on 160-plus pages of specification language.
- Using onboard headers and switches to confirm an understanding of hardware and software operation while finalizing design plans.
- Experimenting with reference design hardware and software

options, while finalizing decisions about mapping the reference design onto a custom board.

- Using the bench top shelf manager during bringup of the custom board, possibly even cabled into a development shelf.

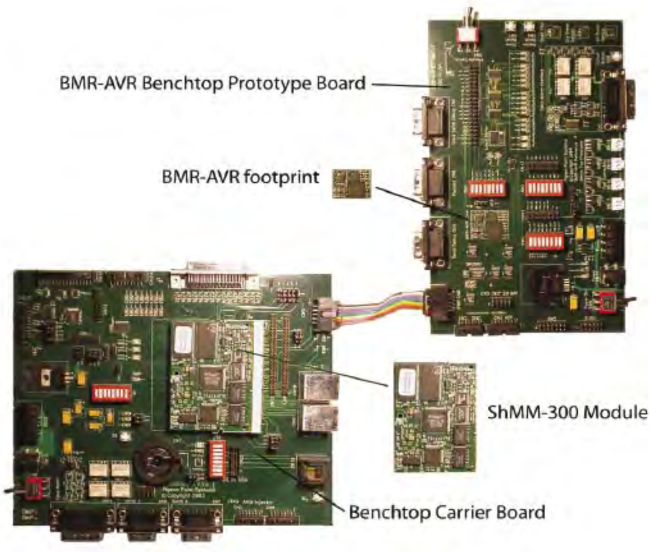

Figure 5

Conclusion

The quickest way to a compliant and interoperable IPM Controller subsystem for an AdvancedTCA board is to use off-the-shelf components. Experts in PICMG management developed and back the IPM Sentry family of components, which includes schematics, source code, bench top hardware, and comprehensive documentation. Using these components can remove the management subsystem of a board entirely from the critical path, allowing developers to focus their energies 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). The current PPS product focus is the IPM Sentry line of platform management components, including AdvancedTCA shelf- and board-level management components. Previously Mark was vice president of engineering at Lynx Real-Time Systems (a UNIX-compatible RTOS supplier) and TeleSoft (a major supplier of embedded development solutions for Ada). 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
Email: mark@pigeonpoint.com
Website: www.pigeonpoint.com

