HIGH AVAILABILITY

Remote, reliable firmware upgrade on PICMG board management controllers

By Mark Overgaard

Intelligent Field Replaceable Units (FRUs) in AdvancedTCA and CompactPCI/CompactTCA include a management controller (an IPM Controller, in AdvancedTCA specification terms) that provides a shelf or chassis manager visibility to inventory data, sensor readings, and similar platform management information. The stringent minimum requirements for these board management subsystems include compliance with the hundreds of pages of requirements in the relevant PICMG and Intelligent Platform Management Interface (IPMI) specifications. Also required is active interoperability testing, including at the AdvancedTCA/AdvancedMC Interoperability Workshops (AIWs) that PICMG organizes several times a year.

In addition to these minimum requirements, firmware in highly available systems must be reliably upgradeable without a physical visit to the equipment. Reliable upgrade typically implies that a separate copy of the upgraded firmware is downloaded to the controller in such a way that the original firmware is preserved. If anything goes wrong at any time during the upgrade process, the original firmware can be restored and the IPM Controller can go back to using it while the difficulties in the upgrade are sorted out.

This article describes how these requirements can be met in off-the-shelf PICMG board management building blocks, such as the IPM Sentry Board Management Reference (BMR) solutions from Pigeon Point Systems.

The IPM Sentry solutions are widely used to reduce the effort to build compliant and interoperable IPM Controllers on boards governed by PICMG specifications. Using an off-the-shelf solution for the IPM Controllers allows board developers to focus their often scarce development resources on the value-add portions of their boards. There are two BMR variants used in new AdvancedTCA designs: BMR-AVR-ATCA (based on Atmel AVR microcontrollers) and the newly available BMR-H8S-ATCA (based on Renesas H8S microcontrollers). These IPM Sentry reference designs are used as examples in this article.

The focus here is on AdvancedTCA IPM Controllers, but similar considerations apply to CompactPCI/CompactTCA IPM Controllers. In addition, though the emphasis here is on IPM Controllers on full-sized boards, firmware upgrading is just as applicable to the IPM Controllers on any other intelligent FRUs, such as fan trays or power entry modules. This discussion also applies to the Carrier IPM Controllers (Carrier IPMCs) and Module Management Controllers (MMCs) on carriers and hot-swappable modules based on the just adopted Advanced Mezzanine Card (AMC) Base Specification, PICMG AMC.0.

Firmware upgrade interfaces

Figure 1 shows the key interfaces of an AdvancedTCA board that are relevant to firmware upgrades for the IPM Controller:

- IPMB-0, the dual-redundant Intelligent Platform Management Bus that links all intelligent FRUs in a shelf to the shelf manager: This standardized interface is present on all IPM Controllers and is accessible (typically over Ethernet) via the shelf manager. Given these properties, this interface is very attractive for remote firmware upgrades.
- Payload Interface to the *payload* or primary function of an AdvancedTCA board, which may include one or more powerful general purpose processors based on the Pentium, PowerPC, or SPARC architectures: This interface is implementation-dependent in AdvancedTCA (and uses UART-based serial ports in the IPM Sentry BMR designs, for instance). Nevertheless, some developers prefer to have a payload processor retrieve firmware upgrade images over one of AdvancedTCA's high-speed interfaces (the Ethernet based Base or SERDES based Fabric Interfaces) and supervise the upgrade process.
- Serial Debug Interface intended for debug or craft person interactions with the IPM Controller: This interface is present on most IPM Controllers, but is completely unaddressed by the AdvancedTCA specification. On IPM Sentry BMR designs, this is a UART-based interface that uses the same Serial Interface Protocol Lite (SIPL) protocol as the Payload Interface. Typically, this interface is not easily accessible on a remote basis, so most firmware upgrades using this interface require a person to be present at the equipment.

The main IPM Controller block in Figure 1 is not drawn to scale, but the inset shows the approximate scaled footprint of the BMR-AVR-ATCA implementation on an AdvancedTCA board.

Overall upgrade process

The Intelligent Platform Management Interface (IPMI) specification, on which the PICMG platform management architecture is based, reserves a set of command codes for firmware upgrades, but does not define any specific command code points or associated semantics. Therefore, the firmware upgrade protocol by which commands using these codes implement an upgrade is currently specific to the developer of the IPM Controller firmware. It would clearly be attractive to have a universal upgrade protocol for all PICMG-governed IPM Controllers. Standardizing such a protocol is a candidate activity in the recently launched effort to update the AdvancedTCA specification.

The standardized protocol would likely emphasize upgrades via IPMB-0, the only fully standardized interface to a PICMG IPM Controller. Upgrade protocol packets would be forwarded from, and corresponding responses forwarded back to, an upgrade facility that either resides with or remotely accesses the shelf manager. Remotely originated upgrade commands could be issued via the shelf manager using the *Send Message with response tracking* forwarding facility defined by IPMI. Figure 2 shows this process. Figure 2 also shows how upgrade protocol packets can reach the MMC on an AMC, using the Carrier IPMC as a proxy and leveraging IPMI's *Send Message with response tracking* mechanism for each packet. For remotely originated packets, the Carrier IPMC would act as a second level proxy, following the shelf manager's first level proxy role.

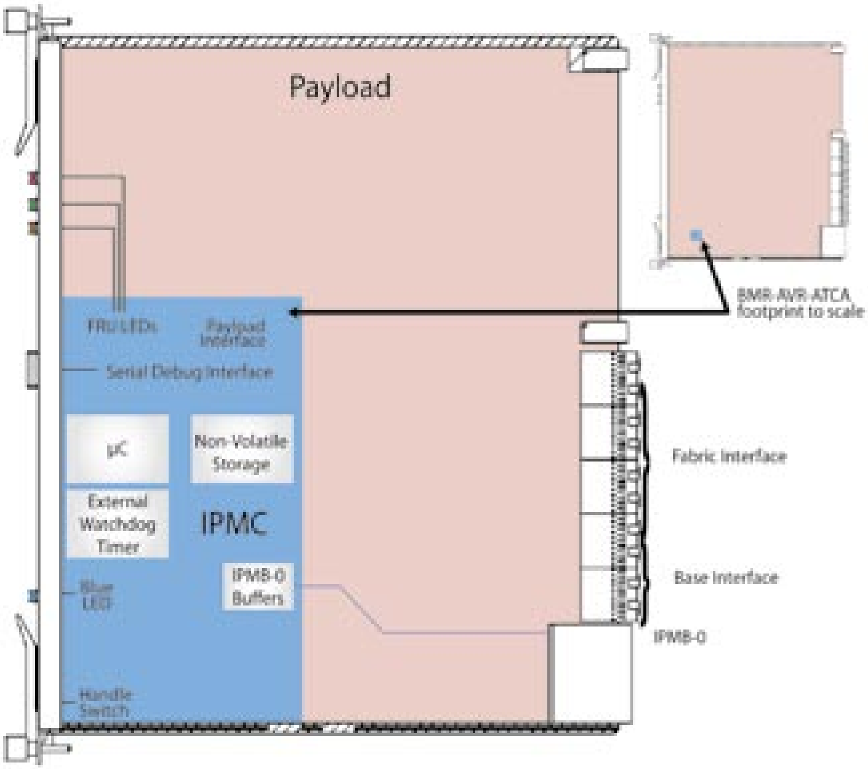

Figure 1

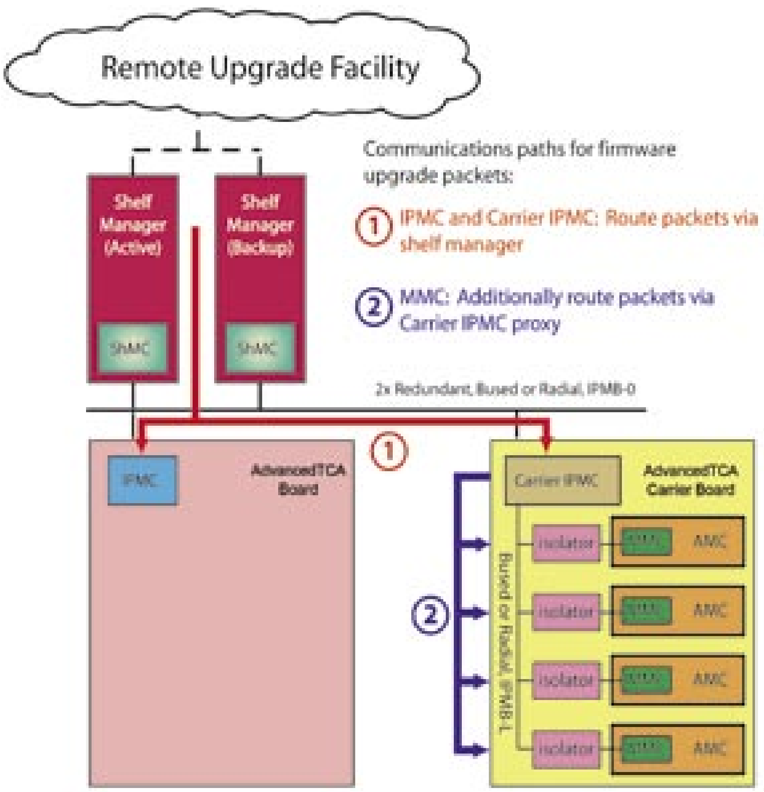

Figure 2

A standardized upgrade facility might also include a format for the upgrade images that are passed to the IPM Controllers being upgraded. Figure 3 shows the format used in the IPM Sentry upgrade facility. It is intentionally designed to be useable as a standardized format. The first three bytes contain an organization identifier (see www.iana.org) to identify the group responsible for the syntax and semantics of the image, followed by a version number for that definition. After this header, the image format is a sequence of N individual images (say for the master and slave AVRs, N=2 in a BMR-AVR-ATCA based IPM Controller). The current IPM Sentry format uses the Pigeon Point Systems identifier in the header. A PICMG-defined format could use the PICMG identifier, just like the PICMG-defined IPMI command extensions do.

With a standardized image format, creation of the image would be the only step specific to vendor or development tool. The upgrade facility of Figure 2 could then deliver that image to an IPM Controller via the shelf manager.

The IPM Sentry BMR firmware upgrade process uses the `mkupgimg` and `upgradefw` utilities to create and download (respectively) the upgrade images.

Storing active/backup firmware copies on IPM controller

A key requirement of reliable firmware upgrade is the storage of two copies of the firmware: one for active use and one as a backup. During the BMR upgrade process, the current firmware is copied to the backup region. Next, the upgrade protocol copies a new firmware image into the active region. If the upgrade process fails or the new firmware image becomes corrupted for any reason, copying the backup copy into the active region restores normal operation. Subsequently, the upgrade process can be retried if necessary. One example of a problem that can occur during the upgrade process is a power failure.

Figure 4 shows how the available Flash memory is allocated for these purposes on the BMR-AVR-ATCA and BMR-H8S-ATCA IPM Controllers. For the BMR-AVR design, the IPM Controller contains two AVR controllers: the master and slave AVRs. The master AVR stores backup copies of both its own and the slave AVR's firmware. The master and slave AVRs assumed in the figure are the ATmega128, with 128 Kbytes of Flash, and the ATmega8, with 8 Kbytes of Flash, respectively. The H8S/2168 controller assumed in the figure has 256 Kbytes of user Flash.

Upgrade protocol

In the absence of a standardized protocol, this article uses the IPM Sentry BMR upgrade protocol to show how reliable remote upgrades can be implemented. A key concept of this protocol is that an IPM Controller is either in upgrade mode or in normal operations mode. If in upgrade mode, the IPM Controller suspends its normal operation and focuses solely on executing the upgrade protocol. Any other commands received in upgrade mode yield the IPMI-defined response: Device in Firmware Update Mode.

The key IPMI commands in this protocol are:

- Firmware upgrade status: Allows an upgrade utility or other entity to query an IPM Controller about its upgrade status (that is, whether it is in upgrade mode).
- Firmware upgrade start: Switches the IPM Controller to upgrade mode.
- Firmware upgrade prepare: Prepares the IPM Controller for upgrading, including copying the current firmware image to the backup region in Flash.
- Firmware upgrade write: Programs a portion of the new image.
- Firmware upgrade complete: Finalizes the new image, exits upgrade mode, and boots the newly programmed firmware.
- Firmware upgrade restore backup: Causes the boot loader to restore the firmware from the backup image.
- Firmware upgrade backup revision: Returns the revision of the backup firmware image(s) currently stored in the IPM Controller.

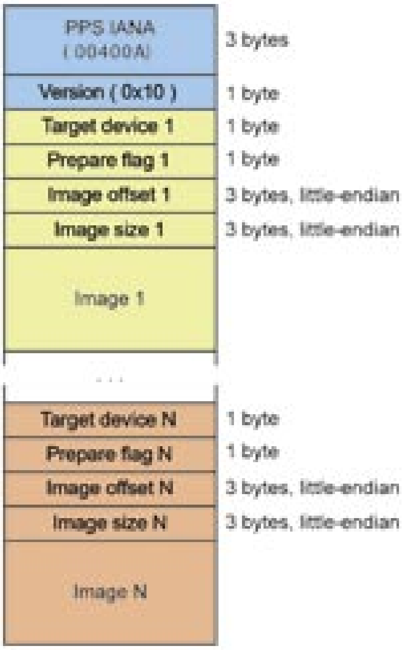

Figure 3

Putting it all together

Provisions for reliable remote upgrades of PICMG management controllers are only a small part of producing the compliant and interoperable controllers on which the rapidly growing AdvancedTCA ecosystem depends. Between the base specifications for AdvancedTCA and AMC, there are 230 pages of platform management recommendations and requirements in addition to the IPMI specification's 430 pages. Thoroughly understanding and developing compliant implementations of these requirements is a serious engineering effort.

It would clearly be attractive to have a universal upgrade protocol for all PICMG-governed IPM Controllers.

Another critical part of producing viable management controllers is testing, to be sure that compliant but independently implemented controllers can interoperate successfully. Crucial for such testing are the PICMG-organized AIWs. These events, held several times a year, bring together dozens of PICMG members for

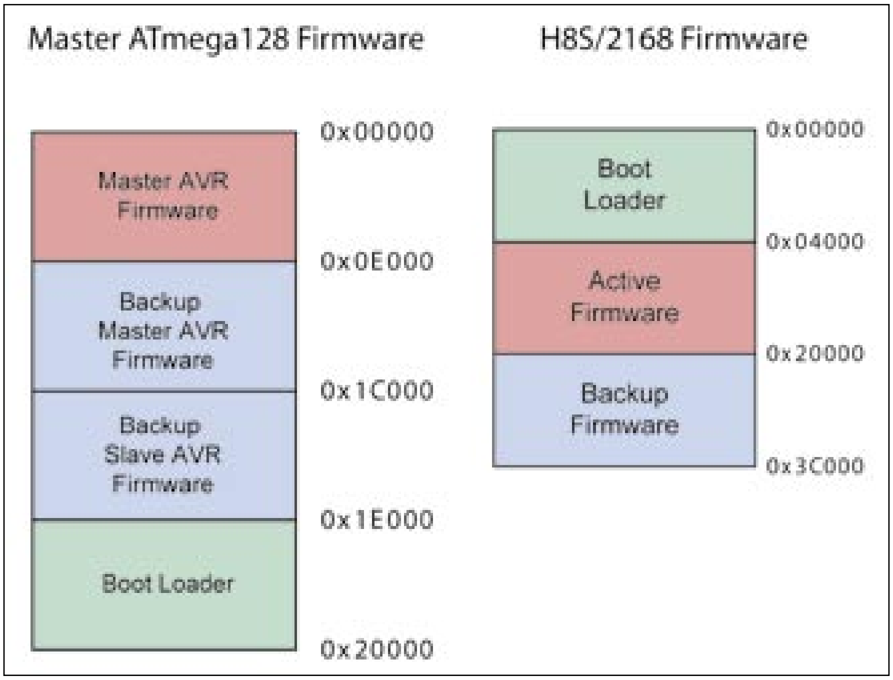

Figure 4

systematic interoperability testing, guided by dozens of test plans covering different functional areas of the AdvancedTCA and AMC specifications.

For board developers who want to focus their engineering efforts on the unique value adds of their boards, versus dealing with all the challenges noted earlier on the microcontroller front, an off-the-shelf management solution can be a significant win. The IPM Sentry family of Board Management Reference solutions provides comprehensive design specifications, tested schematics, ready-to-use bench top implementations, highly configurable firmware source code, and integrated firmware development tools.

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 M.S. in Computer Science from UC San Diego and a B.S. 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
Web site: www.pigeonpoint.com