Embedded BIOS®

with StrongFrame® Technology

Product Brief

Embedded Tools GmbH
Willy-Brandt-Weg 33
48155 Münster
Fon: +49 251 98729-0
Fax: +49 251 98729-20
www.embedded-tools.de

Table of Contents

Business Fundamentals

- 6th Evolution of mature, widely-deployed code base. Deployed on more diverse targets than any other desktop BIOS kit, Embedded BIOS has run on 80186 microcontrollers all the way up to systems involving tens of thousands of 64-bit blades—all with the same mature, proven code base. Not just a BIOS for desktops, Embedded BIOS with StrongFrame Technology(EBSF) is a BIOS for the entire X86 spectrum.
- Most flexible and configurable BIOS SDK available. Over 1,000 simple source-level build options, descriptive tables, and ~90 OEM-configurable callouts, all optional, ensure OEM policies take precedence over industry norms for precise, dependable behavior.
- Fastest, lowest-risk BIOS deployment system in industry makes BIOS a solution, not a problem.
- Deep compatibility with 20 years of X86 industry standards and OSes for support of the industry's most mature software development tools and building blocks.
- Widest support for x86 chipsets, CPUs, and SIO building blocks keeps customer roadmap options open and flexible.
- Architectural support for multiple minor releases within same powerful EBSF development framework eliminates overhead and risk associated with moving to new minor releases.
- Architectural support for multiple releases of chipset, CPU, SIO, and board adaptations provides full-lifecycle support for customer's entire SKU set within a single development tree.

Technical Fundamentals

- Full source code for core BIOS SDK creates a perfect fit build, every time. Keeps environment open; no hidden "black magic" in the core. Straight-up, highly-readable, well-documented and organized.
- Supports Microsoft Visual Studio with MASM 8 and above, including Visual Studio Express, the free version made available by Microsoft on the web; as well as both Intel and Microsoft ASL compilers, also available on the web. No antiquated or unavailable tools and no extra costs.
- Off-the-shelf add-on modules support north and south bridges, processors, and SIO modules from major silicon manufacturers. These modules typically perform DRAM configuration, cache control, PCI Express, and other highly technical silicon-specific functions. Customers use these modules as building blocks, never needing to modify them, just as they never need to modify the BIOS core.
- Full-source working sample board-level modules demonstrate working BIOS for silicon manufacturer development boards. Useful as springboards to accelerate board bring-up, boot to DOS, and the adaptation steps that follow. Extensible to introduce proprietary pre-boot policy or even run firmware applications before the OS loads.
- Runs highly-optimized 32-bit Firmbase Technology V2 environment with High Availability stack, TCP stack, USB stack, and framework for running OEM-written SMM applications built with 32-bit Windows PE tools. Uses as little as 50us SMM latency (hardware and OEM configuration dependent.) SMM environment runs concurrently with foreground's interrupt-driven serial

communications to 115kbaud (USB mass storage and some OEM Firmware Applications may, for performance reasons, exhibit additional latency when performing I/O.)

- USB stack supports OHCI, UHCI, and EHCI (USB 2.0); up to 8 USB mice and 8 USB keyboards; up to 8 CD/DVD, 16 hard disks, 4 floppy disks, and other USB storage devices; complex hub topologies. (Boot from USB sold separately.)
- Move Embedded BIOS 2000 V5 adaptations to the newest StrongFrame Technology core in as little as a week; two weeks typical. Conversion mostly involves removing BPM assembly code associated with ACPI, PCI, MP, and Setup, and replacing with descriptive tables, leaving the heavy lifting to the core, not the BIOS deployment engineer. Project files reduced by ½ to one-tenth of their original size.
- BBS feature adds automatic boot device configuration and prioritization (including boot ROMs such as PXE or SCSI cards, mass storage devices such as ATA, IDE, USB and ATAPI floppies, hard disks, and CDROMs.) Configurable to "Boot from PCI slot X", "Boot USB hard drive", "Boot USB floppy", "Boot any bootable disk", "Boot to debugger", "Boot Windows CE", "Boot EFI", and much more.
- Most-configurable and flexible PC-Based Setup system. Entirely table-driven, no assembly language required. Includes context-sensitive help at field-level, multiple core and OEM defined menus, dependent fields, field security, real-time fields (including bar graphs), support for different screen shapes and sizes and color sets, console redirection over serial in color using standard, widely-available terminal emulators such as HyperTerminal, and storage to CMOS and OEM-defined media (which could include Flash, TPM, or other devices.)
- Integrated Core Diagnostics Framework provides burn-in diagnostics and makes it easy for OEMs to add diagnostics suites to the system BIOS.
- OEM-expandable Preboot Applications menu with integrated user interface gives unified look and feel to Setup, BBS menu, integrated diagnostics, and OEM-written applications; simplifies code and allows applications to run over serial console redirection. Add an IP-based KVM for network-based consoles.
- Configure ACPI, PCI, MP, and Setup with tables that describe the platform instead of writing assembly code, eliminating thousands of lines of assembly code that would otherwise need to be designed, produced, and maintained.
- Powerful 2nd-generation compression in core shrinks system BIOS and modules on the ROM; expands them during POST into RAM saving boot ROM real estate and BOM costs, allows developers to include high-end features like HTTP server, TCP stack, Security stack, High Availability tools, and OEM-written C code that runs in the Firmbase environment. Up to 20% higher compression over V5 core.
- Embedded PCI option ROMs automatically expand and run when devices detected during POST, saving ROM space and simplifying deployment. Embedded option ROMs individually replaceable in the field with Reflash utility.
- PCI configuration supports large I/O topologies including high-end bridges and 64-bit PCI address space.

Summary Description

Embedded BIOS® with StrongFrame® Technology (EBSF) is the 6th evolution of Phoenix Technologies' firmware SDK solution for x86 designs, offering firmware deployment designers unsurpassed configuration flexibility, rapid and risk-reducing deployment tools, deep compatibility with x86 industry standards, support for the widest range of embedded chipsets, CPUs, and SIO building blocks, and support for multiple core versions and module versions within the same architectural framework.

Mature, Widely-Deployed Code Base

The industry's leading solution to building BIOS firmware for the widest range of OEM equipment, Embedded BIOS has been deployed over the entire range of embedded verticals (datacom/telecom, industrial automation, consumer electronics, office/information automation, military/aerospace, automotive, medical, gaming, retail automation), and IT infrastructure computing platforms (from pocket PCs, laptops and notebooks to 32- and 64-bit servers, blades, and clusters beyond 10,000 nodes.) For over a decade, Embedded BIOS has built perfect fit BIOSes in terms of functionality, size, and performance for the entire range of x86 platforms, and will continue for years to come.

The Most Flexible and Configurable BIOS SDK

With over 1,000 simple source-level configuration options, Embedded BIOS is the most configurable BIOS available today. But that's only the beginning, because the EBSF core's major features are configured with intuitive descriptions that the OEM customer uses to tell the core how to build ACPI, MP, PCI, and Setup support automatically, eliminating thousands of lines of otherwise tediously-written assembly language.

Beyond configuration of the core and describing the hardware itself, the OEM customer can also supply a Board Personality Module (BPM) that can intercept nearly 90 calls from the core to influence virtually all core decisions and perform special actions during POST, precisely controlling platform behavior.

Multi-core and support module versioning architecture built into the EBSF framework extend the flexibility of the SDK to allow different firmware projects to be validated with different levels of the core itself, as well as with different versions of chipset, CPU, SIO, and other modules. This multi-versioned framework facilitates Phoenix Technologies' rapid distribution of its core innovations as well as chipset manufacturer errata to OEM customers without requiring that each project within that framework use the most current version. Customers may, for example, wish to leave some projects at earlier levels, depending on where they are in their lifecycle. Multi-core and support module versioning make it possible to maintain all firmware projects within one centralized development tree that maximizes flexibility and code reuse, while keeping all builds at the customer's fingertips.

Enabling Rapid, Low-Risk Deployment

With Embedded BIOS, BIOS developers can produce working prototypes in days, and finish deployments in weeks, thanks to the EBSF core's new configuration management based on platform description using intuitive and descriptive methods, rather than assembly language. While assembly code is still used for fine detailed control over the configuration, objects like ACPI, MP, and PCI tables are automatically constructed by the core, as are Setup screens and their fields, based on descriptions of the target supplied by the developer. With less assembly code to write, OEM customers can achieve measurable results much earlier in the development process, while at the same time reducing the amount of code that must be validated in system test.

Broad Range of Silicon Support

OEM customers using Embedded BIOS enjoy rapid, low-risk deployment to the broadest range of the x86 industry's chipsets, CPUs, and SIO packages. Unlike IT BIOS companies that focus only on IT-based chipsets used by 18-20 1st-tier IT ODMs, Phoenix Technologies' larger and more diverse OEM customer base requires support for a much broader range of board designs and silicon, including legacy chipsets to support long-life supply chains, and the most cutting edge hardware technologies only just becoming

available. OEM customers gain the flexibility to move quickly to different silicon platforms as business and technical needs evolve.

An Important Part of Every Firmware Project

Embedded BIOS empowers OEM makers of X86-based computing equipment to take control over the firmware layer in every project, reducing risk-laden dependencies on outside vendors, increasing economy of scale, and building-in value-add features across all platforms. OEM customers may elect to have Phoenix Technologies use the SDK to produce a turn-key BIOS, or they may use Phoenix Technologies' consulting and engineering services to augment an existing in-house BIOS team to get a jump start on their project and optimize familiarity with the SDK, or they may take advantage of Phoenix Technologies' training programs to get up to speed quickly before performing their own in-house adaptation using the SDK. The Embedded BIOS SDK is productized with these options in mind, to accommodate a diverse range of embedded OEM customers' needs.

Deep Compatibility with 25 Years of Industry Standards

Firmware in other processor architectures (i.e., PowerPC, ARM, MIPS) often takes the form of a BSP within a kernel's framework, but it is that integration with the OS that is often its limitation. The x86 architecture is open in a way that other architectures are unlikely to achieve for years to come, because it has so much history behind it. To truly be compatible with today's PC architecture, a BIOS must offer all the external APIs and internal nuances of the IBM PC, XT, and AT BIOS implementations, as well as support the industry initiatives and operating systems that were introduced since then. Embedded BIOS does just that, building on a decade of support for these initiatives on the widest possible range of chipsets, CPUs, and board designs.

Summary List of Industry Standards Supported by Embedded BIOS (not inclusive)

- ACPI 2.0 Specification
- AGP 3.0 Specification
- AHCI Specification (dependent on chipset/platform)
- APIC/IOAPIC
- APM 1.2 Specification
- ATA6 Specification
- ATAPI 1.0 Specification
- BBS 1.01 Specification
- BIOS32 Specification
- DMI/SMBIOS 2.3.1 Specification
- EFI 1.0 Specification #
- Enhanced Disk Drive 3.0 Specification
- El Torito 1.0 CD-ROM Boot Specification
- Firmbase 2.0 Specification
- JEDEC DRAM Specifications (dependent on chipset/platform)
- Legacy Free
- LPC 1.1 Specification
- Microcode Update
- Microsoft Debug Port 1.0 Specification
- Microsoft Simple Boot Flag 1.0 Specification
- MP 1.4 Specification
- PAE to 64GB
- PCI 2.2 Specification
- PCI Express 1.0A Specification
- PCI-X 2.0 Specification
- PE 6.0 Specification (for Portable Executables loaded by BIOS)
- PMM 1.01 Specification
- PnP 1.0A Specification

- PXE Specification*
- System Management Bus BIOS Interface 1.0 Specification
- UDMA6 Specification
- USB 1.0, 1.1, 2.0 Specifications, including Legacy USB and Boot from USB

Value-Add Features

Embedded BIOS goes beyond reduced risk, rapid deployment, or compatibility with standards. Rolled into the powerful EBSF core is a wealth of features that may be selectively enabled and tuned on a project basis by OEM customers.

Summary List of Special Core Features in Embedded BIOS (not inclusive):

- Boot from PATA/SATA (including native mode), ATAPI, IDE, Iomega ZIP, LS-120, SCSI storage
- Boot from USB CDROM, hard disk, floppy, disk on key, and other mass storage devices *
- Boot from network, PCI slots, ROM images, Windows CE, and EFI
- Boot to debugger, preboot menu, setup screen, or OEM-written code
- Setup configuration stored in CMOS or OEM-designated alternate devices
- CMOS backup/restore from Flash
- Console redirection over serial link
- Common User Interface (CUI) menuing system supports preboot applications built into BIOS
- Debugger (BIOS-aware)
- Driver architecture for BIOS modules
- Disk emulators using ROM, RAM, and Flash
- EFI loader
- Embedded option ROM extensions
- Firmbase Technology, including:
    - Virtualized emulation of PCI, I/O, and memory-mapped hardware with OEM-written code
    - USB stack (OHCI, UHCI, EHCI, USBD, USBMS and USBHID)
    - TCP stack (MACs, ICMP, DHCP, IP, UDP, TCP, TFTP, Telnet, HTTP, SMTP, SNMP)
    - Web server in firmware can respond to maintenance requests even if OS crashes
    - Email client in firmware can send email based on OEM policies even if OS crashes
    - Trusted Computing Base (TCB)
    - Multi-user command shell
    - High Availability (HA) subsystem
        - HA Monitor monitors entire system health, implements OEM recovery policies *
        - Hardware Monitor monitors hardware health with OEM-defined plugins *
        - OS Monitor monitors OS health including processes (Windows and Linux) *
    - Platform Update Facility *
    - Boot Security Application *
- Flash programming API, exposed through INT 15h to BIOS and DOS applications
- Graphical desktop UI. Can boot DOS in window, Firmbase in window. API exposed to OEM firmware applications. GUI up and running in as little as 500ms (hardware and platform-dependent.)
- Headless operation
- HTML browser BIOS user interface
- POST codes to any I/O port
- POST diagnostics
- POST memory tests including quick, standard, and exhaustive options in low and high ranges
- Preboot menu and applications
- Preboot diagnostics suite
- Preboot manufacturing mode server
- Preboot system monitor (real-time target analysis)
- Preboot system configuration browser
- Preboot registration screen system

- Preboot user login/logout
- Preboot virtual consoles provide access to Firmbase shells and log
- Quick Boot with sub-second POST (dependent on hardware platform and other factors.) Fastest in industry.
- Reflash BIOS ROM in-circuit
- RTOS loader
- User-level security based-on Firmbase Technology's Trusted Computing Base (TCB). Users assigned individual access controls for system objects; protected objects include setup menus and fields, boot actions, and OEM-defined actions. TCB supports plug-in Security Authorities allowing authentication and authorization using any OEM-defined means.
- Setup engine, data driven, with real time fields, security, OEM-customizable to any extent. Standard menus include the following (but can be removed, replaced, or augmented by OEM):
    - Boot devices (BBS), IDE, and Floppy configuration
    - Chipset configuration
    - Exit menu
    - Core features (ACPI, APM, MP, Quick Boot, etc.)
    - Firmbase configuration
    - Main menu (system identification, RTC setup)
    - Miscellaneous configuration
    - PnP configuration
    - POST configuration (error handling, headless, etc.)
    - Security configuration
    - Shadowing configuration
    - User interface configuration (colors, etc.)

- Splash screen supporting graphics, animation, and sound
- Watchdog timer support
- Windows CE and EFI loaders

* SDK option, sold separately
# Requires Intel EFI toolkit to build EFI load image

Embedded Products
915 118th Ave. SE, Suite 320, Bellevue, WA 98005, 800-850-5755, 425-576-8300
www.phoenix.com, embedded_sales@phoenix.com