

EMBEDDED IDE FOR COLDFIRE PROCESSORS

HIGHLIGHTS

- Complete gcc ColdFire Toolchain
- Full GUI Debugger
- GUI Project Manager
- Integrated VI editor
- Integrated GREP Utility
- Complete gdb-to-Target Communication Support
- Entire ColdFire Family Supported, including v4e
- Simple Subscription License
- Comprehensive Runtime Library

Blunk Microsystems provides development tools, system software, device drivers, and board support packages to the embedded systems market.



Software development tools from Blunk Microsystems, TargetTools™, provides an Integrated Embedded Development Environment (IDE) for ARM, ColdFire, and PowerPC processors. TargetTools delivers the features developers need to ensure on-time development and delivery of their embedded projects.

TargetTools™ includes the following features:

- Windows port of entire GCC toolchain, including compiler, assembler, linker, and loader. Uses MinGW to eliminate problems from conflicting CygWin DLL installations and to provide faster compilation.
- Windows port of Insight/GDB. Full GUI debugger that provides excellent performance and reliability. Supports stack trace, register, breakpoint, expression watch, memory, local variable, source, and console windows. The main window can be switched between the following displays: source, assembly, split source-assembly, and mixed source-assembly. Includes extensive online help.
- Three forms of kernel awareness: Object Status window shows state of tasks, semaphores, queues, etc. Kernel Event Trace window shows “software logic analyzer” view of context switches, interrupts, and service calls versus time, with ability to scroll and zoom. Clicking a line in the task list window scopes the source code window to the current execution point of the selected task.
- gBuild™ Project Manager. Add or delete project files using graphical user interface. Projects are classified as either libraries or applications. A single button click automatically checks file dependencies and time stamps, and compiles, assembles, and links files as necessary to rebuild the project. Optionally launches the editor at appropriate line when a compiler error occurs.
- On-Chip debug agent allows GDB to communicate with the target board. ‘Talks’ GDB’s remote debug protocol and calls on-chip driver to control the target processor. Supports Abatron’s BDI2000 BDM and JTAG interfaces. Supports P&E Microcomputer Systems’ Ethernet, parallel port, PCI, and USB BDM interfaces.
- TargetMon™. An advanced debug monitor that communicates with GDB over Ethernet. Provides extremely fast download times and responsiveness to debugger commands. In addition to debugging task mode code, TargetMon supports placing breakpoints in, and stepping through, interrupt service routines.
- Supports all members of the ColdFire family, from low-end MCF52XX parts to high-end MCF547X/8X parts using GCC version v4.1.1. Includes direct support for the new instructions introduced in the v4e core used in the high end MCF547X/8X processors.

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multitasking, priorit based, preemptive kernel. Includes Standard C library.

File Systems

High performance file systems for Flash, HD, and RAM. Implements power fail recovery, access protection, links, quotas, and other advanced features. Supports POSIX API.

TargetTCP

High Performance TCP/IP Stack with Standard Sockets API.

TargetWeb

Embedded web server allows monitoring (data output) and control (data input) via desktop browser. Requires TargetTCP and any POSIX file system.

- His GrepShip™. A Windows utility for finding or replacing text strings in multiple files. Easy to use and fast. Supports regular expressions and wildcard file names. Optionally searches subdirectories. Double clicking a line in the match window launches the editor on the associated file. Optionally waits for confirmation during replacements.
- Lemmy™. A full featured, easy to use port of the popular VI editor. Excellent Windows integration. Supports multiple split windows into the same file, many configuration options, syntax highlighting, and double click selection of search text. Can launch His GrepShip for a multiple file search using the current search text.
- Offered as a subscription ltcense allowing unlimited use on a single PC during the subscription term. Comes with full technical support including free access to bug fix and upgrade releases during the subscription period and answers to technical questions via email, phone, or fax.
- Delivered as a standard Windows installer, installation is completely turnkey and includes an 'uninstall' option. A 'TargetTools' submenu is added to the Windows 'Program' menu and registry values are used to store configuration options. TargetTools is a completely Windows integrated GUI environment.
- Integrated with PC-Lint from Gimpel Software. Apply the power of this award-winning static analysis program to your project files via a simple GUI interface. Supports unit testing of single or selected files as well as 'deep lints' that include every project file.
- TargetTools updates are installed without overwriting or corrupting previous compiler or debugger components. Multiple versions can be installed or/and deleted at any time. Switching between versions is easily done using GUI selections for 'Version' and 'CPU'. This supports regression testing and low risk adoption of updates.
- Blunk Microsystems provides a range of other high performance, royalty free embedded software modules that drop into the TargetTools environment without porting. These include **TargetFFS**, an embedded flash file system, **TargetTCP**, an embedded TCP/IP protocol stack, and **TargetWeb**, an embedded Web server. ■

Licensing Terms

TargetTools™/ColdFire is offered as a subscription license that allows unlimited use during the subscription term on a single PC. Includes full technical support during the subscription period.

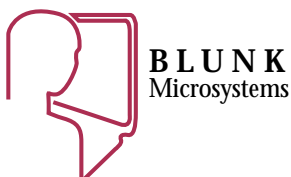
TargetOS™

REAL-TIME OPERATING SYSTEM

HIGHLIGHTS

- Royalty Free
- Multi-Tasking
- Provides Semaphores, Queues, Events
- Extensive Timer Support
- Deterministic and Responsive
- Flexible Scheduler
- Small Footprint
- CodeWarrior Kernel Awareness
- Includes Standard C Library
- Command Line Monitor

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



MULTI-TASKING

TargetOS is a real-time, priority-based, preemptive kernel. The scheduler shares the CPU among various tasks. Tasks are lightweight threads having a common address space and a single set of file descriptors. Tasks can be dynamically created and deleted and are always in one of five states: running, ready, blocked, suspended, or delayed. Only one task at a time is the running task.

When preempted by a higher priority task, the running task is placed at the front of the ready list for its priority level. Thus when the higher priority task blocks, the preempted task again becomes the running task. When the running task blocks to wait for an event or resource, it is placed at the end of its ready list.

Task priorities are assigned at creation and can be dynamically changed. Priority changes take effect immediately with tasks pending in a priority queue dynamically re-ordered in the queue. Priority inheritance is implemented to prevent priority inversion. This temporarily boosts the priority of any task owning a mutex that is being requested by a higher priority task.

SUPPORTS TASK COMMUNICATION

An application partitioned into multiple tasks requires support for task communication and synchronization. TargetOS provides event groups, message queues, counting semaphores, and a task sleep/wake mechanism. Both tasks and interrupt service routines can post events, messages, or semaphore tokens.

- Semaphores and queues perform either FIFO or priority scheduling of pending tasks.

- Queues are either limited to a maximum number of backlogged messages or allowed to grow indefinitely.

- Mutexes are binary semaphores with the addition of ownership and priority inheritance and are used to prevent priority inversion when tasks with multiple priorities use the same resource.

- Multiple tasks can block on a single event and a single event can make multiple tasks ready. Tasks wait for events using a 32-bit mask to select the events they respond to.

- Tasks sleep either indefinitely or for a number of kernel ticks. Tasks become ready after either timing out or a wake call from another task or interrupt service routine.

DETERMINISTIC AND RESPONSIVE

Interrupt latency and the execution time of most system calls do not increase as the number of tasks, semaphores, queues, etc., increase. The two exceptions, the get-identifier and the delete-object calls, are not usually time-critical. No loops are performed in the kernel with interrupts masked.

Timer-related processing that is variant, such as sorting the timer queue, is performed using background tasks. This allows the CPU to return more quickly to application tasks when a service call requires timer processing.

FLEXIBLE SCHEDULER

Preemption can be permanently disabled at compile-time, in which case each running task keeps the CPU until making a call that blocks. Preemption can be temporarily disabled at run-time by calling `taskLock()`. Preemption

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

OPTIONAL COMPONENTS

TargetTCP

RFC compliant TCP/IP protocol stack providing the standard Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system. Implements wear-leveling to prolong life of flash media. File system integrity is guaranteed across unexpected resets.

TargetLAPB

ISO/IEC 7776 protocol stack. Supports exchanging data on point-to-point networks. Provides automatic flow control and data reliability.

remains disabled until the running task either calls `taskUnlock()` or makes a call that blocks.

Round-robin scheduling among tasks of equal priority is supported. `OsTicksPerSlice` determines how long a round-robin task keeps the CPU if tasks of equal priority are ready. After the specified number of ticks elapse, the running task is placed at the end of its ready list and the first task in that list becomes the running task.

EXTENSIVE TIMER SUPPORT

Timers can be used in a wide variety of calls. Semaphore tokens, messages, and events can be posted after a delay or posted repeatedly at intervals. User functions can be called after a delay or called repeatedly at intervals. Timers can be stopped, tested for expiration, and/or deleted at any time.

EASY TO USE

Kernel objects (tasks, semaphores, timers, etc.) are created dynamically upon application request. There is no configuration file to edit and maintain. The only limit on the number of kernel objects is the amount of available memory. To help catch errors during development, a compile-time option promotes service call parameter errors to fatal errors that break into the command line monitor after printing an error message to `stderr`.

CODEWARRIOR™ KERNEL AWARENESS

The kernel-awareness DLL lists every task and can change the debugger's source code display to show the execution point of any selected task. The state of tasks, semaphores, etc. can be displayed as well as a trace of recent kernel events: system calls, context switches, and interrupt service routines.

INTERRUPT SERVICE ROUTINE MANAGER

TargetOS allows interrupt service routines to be written entirely in C. Assembly language "wrapper" routines save and restore the necessary CPU registers. Interrupt nesting is fully supported. The CPU vector table is initialized at startup with default values to allow detection of bus errors and stray interrupts.

STANDARD C LIBRARY SUPPORT

The TargetOS Standard C library supplies re-entrant stream I/O and memory allocation routines. UART and real-time clock drivers have documented interfaces to support the "stdio.h" stream I/O and "time.h" routines.

COMMAND LINE MONITOR

The command line monitor displays the state of the kernel objects (tasks, semaphores, etc.) and can be started by entering <CTRL-C> on the TargetOS console. Applications and drivers can extend the monitor with custom commands.

TURNKEY SOLUTION

Provided with full source code, default compiler settings, linker command files, user's manuals, benchmarking application, and one year of technical support. Blunk Microsystems offers board support packages and drivers for a range of commercial CPU boards and provides competitive bids on custom board support packages and drivers. ■

Licensing Terms

TargetOS™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

HIGHLIGHTS

- High Performance
- Robust and RFC Compliant
- Multicasting and IGMPv2 for Low-Overhead Streaming Audio and Video
- Optional SNMPv3 support with Integrated MIB-II Agent
- APIPA for Automatic IP Address Assignment on Local Networks
- Optional Embedded Web Server
- Compact and Configurable
- Built-in Protocol Decoder
- Works with or without an RTOS

Blunk Microsystems provides development tools, system software, device drivers, and board support packages to the embedded systems market.



TargetTCP is a fast, reliable, re-entrant TCP/IP implementation developed and supported by Blunk Microsystems that has been extensively tested and demonstrated interoperable with Windows, UNIX, and other TCP/IP protocol stacks.

RFC-compliant with a full protocol suite. Supports TCP, UDP, IP, ICMP, Multicasting, IGMPv2, ARP, APIPA, Ethernet, PPP (with CHAP, CHAT, and PAP), PPPoE, SMTP, NAT, and Wi-Fi. Includes servers for FTP, DHCP, RARP, Telnet, TFTP, and SNMPv1 (optional SNMPv3). Includes clients for DNS, FTP, RARP, and TFTP.

Standard Berkeley sockets application program interface with the following enhancements:

- `connect()`, `recv()` and `send()` timeouts can be set by `setsockopt()`.
- `MSG_WAITALL` flag makes `recv()` block until all requested bytes have been received.
- `setsockfunc()` installs a socket callback function, supporting event driven programming and allowing a single task to service both socket and other events.
- `setsockopt()` `IP_TOS` and `IP_TTL` options set the IP header type-of-service and time-to-live values on a per socket basis.

High Performance. No data copies are performed within the stack. Matching of IP fragment lists, ARP address records, and TCP sockets is done with a fast hashing function. The TCP and UDP checksum calculation is done in assembly language (for PowerPC, ColdFire, and 68K). Access to the assembly language “add with carry” instruction enables the calculation to be done with fast 32-bit accesses.

Supports RISC processors without requiring an extra data copy within the Ethernet driver to align packets. IP, TCP, and UDP headers are

4-byte aligned within the stack, allowing fast access to the 32-bit fields in these structures. Application data remains unaligned, eliminating an extra copy required by Berkeley-derived implementations.

Using the Berkeley sockets API, only one copy of application data is performed. This applies to inbound and outbound transfers using both UDP datagrams and TCP streams. A zero-copy API is provided which eliminates the copy associated with the sockets API.

Supports unnumbered serial links (RFC1716) to avoid wasted IP addresses and artificial subnets on targets with multiple PPP interfaces.

Supports TCP out-of-band data and urgent data mark. Both socket callback functions and `select()` can be used to notify applications when a connected peer has entered urgent mode.

Easy to administer. The DHCP client may be configured to use gateway and DNS server addresses supplied by the server. The RARP client may be configured to use the RARP server as a default gateway. PPP connections can be configured to be a default gateway and to request a DNS server address from the remote peer.

Integrated. The Telnet server is integrated with a command line monitor that provides a full set of built-in commands, access to the TargetOS™ command line monitor, and easy extensibility. The FTP server uses TargetTCP's zero-copy interface and will serve files using any POSIX compatible file system. The TFTP client is integrated with the TargetOS runtime library, allowing access to remote files via `fopen()`, and with the TargetOS loader, which accepts both ELF and S-record formats.

Includes an SMTP client. Send email from your embedded target using only the SMTP server name or IP address, the sender and recipient email addresses, a subject string, and a ▶

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetWeb

Embedded web server allows monitoring (data output) and control (data input) via desktop browser. Requires TargetTCP and any POSIX file system.

File Systems

High performance file systems for Flash, HD, and RAM. Implements power fail recovery, access protection, links, quotas, and other advanced features. Supports POSIX API.

pointer to the message text. Ideal for sending alerts when a limit is approached or a critical event occurs.

TargetTCP has a clearly documented network driver interface and supports concurrent use of multiple network interfaces. Ethernet and PPP drivers for the MCF5475, MCF5272, MPC860, DP83815, i82559, LAN91C111, and other controllers are available separately. Blunk Microsystems provides competitive bids on custom drivers.

Built-in TCP/IP protocol decoder. When enabled by a compile-time flag, a short summary of every received or transmitted packet is printed to stdout on an ongoing basis. Ideal for testing new drivers or verifying new configurations.

Compact and configurable. Applications typically require approximately 30 KB of code from the TargetTCP library and can be configured to use as little as 32 KB of

RAM data.

Shipped with several sample applications: an SNMP time client, a chargen, discard, and echo client, a chargen, discard, and echo server, and an HTML client that cycles through a list of URLs, downloading each site's home page.

Developed and maintained using TargetOS, Blunk Microsystems' real-time operating system. Easily ported to other real-time kernels or to polling environments that do not use a kernel. A pSOS port is maintained for Philips' Nexperia™ processor.

The TargetTCP source code is 100% ANSI C (other than the optional assembly language checksum routine) and is tested using GCC and CodeWarrior™, the embedded IDE from Metrowerks.

Royalty free. Includes full source code, user's manual, sample applications, and one year of technical support. ■

The TargetTCP Socket Interface:

```
int accept(int socket, void *addr,
            int *addrlen);
    Accepts connections for TCP servers.

int bind(int socket, void *addr,
          int addrlen);
    Attaches a local port number and IP address
    to a socket.

int closesocket(int socket);
    Closes the socket connection, if one exists,
    and recovers socket resources.

int connect(int socket, const void
            *addr, int addrlen);
    Assigns a remote socket address and
    actively initiates TCP connections.

int getpeername(int socket, void
                *addr, int *addrlen);
    Retrieves a socket's remote IP address and
    port number.

int getsockname(int socket, void
                 *addr, int *addrlen);
    Retrieves a socket's local port number and,
    if connected, IP address.

int getsockopt(int socket, int
                level, int optname, void *optval,
                int *optlen);
    Reads various options that control socket
    or protocol behavior.

unsigned long htonl(unsigned long
                    hostlong);
    Converts 32-bit integers from host byte
    order to network byte order.

unsigned short htons(unsigned short
                     hostshort);
    Converts 16-bit integers from host byte
    order to network byte order.

ui32 inet_addr(const char ptr);
    Converts an IP address from a dotted-decimal
    string to a network byte ordered integer.
```

```
int ioctlsocket(int socket, int
                request, void *arg);
    Sets or gets a socket's mode.

int listen(int socket, int backlog);
    Prepares TCP sockets to accept client connections.

unsigned long ntohl(unsigned long
                    netlong);
    Converts 32-bit integers from network order
    to host byte order.

unsigned short ntohs(unsigned short
                     netshort);
    Converts 16-bit integers from network order
    to host byte order.

int recv(int socket, void *buffer,
          int buflen, int flags);
    Receives data from a socket.

int recvfrom(int socket, void
              *buffer, int buflen, int flags,
              void *from, int *fromlen);
    Receives data from a socket.

int select(int fdnum, fd_set *reads,
            fd_set *writes, fd_set *excepts,
            struct timeval *timeout);
    Simultaneously monitors multiple sockets.

int send(int socket, void *buffer,
          int buflen, int flags);
    Sends data over a connected socket.

int sendto(int socket, void *buffer,
            int buflen, int flags, void *to, int
            tolen);
    Sends data over a socket.

int setsockopt(int socket, int level,
                int optname, void *optval, int
                optlen);
    Sets various options that control socket or
    protocol behavior.

int shutdown(int socket, int how);
    Shuts down a socket connection.

int socket(int family, int type, int
            protocol);
    Allocates a TCP or UDP socket.
```

Licensing Terms

TargetTCP™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

HIGHLIGHTS

- **Reliable, Re-entrant Embedded File System**
- **Provides POSIX and Standard C APIs**
- **Guaranteed Integrity Across Unexpected Resets**
- **Use of NAND Flash Memory is Invisible to Applications**
- **Supports Dynamic Creation and Deletion of Files, Directories, and Links**
- **Performs Wear-Leveling, Bad Block Management, and ECC**
- **Supports Multiple Volumes of Unlimited Size**
- **Licensed as Royalty-Free Source Code**
- **Portable to Virtually any RTOS and Toolchain**
- **Includes Sample Applications**

Blunk Microsystems provides development tools, system software, device drivers, and board support packages to the embedded systems market.



TargetFFS-NAND™, Blunk Microsystems' NAND Flash File System, includes the following features:

- A reliable, re-entrant file system with a full POSIX and ANSI C compliant application program interface. Use of NAND flash media for the backing store is invisible to the application layer.
- Supports dynamic creation and deletion of files, directories, and links with full read and write capability. Not a static ROM-image file system.
- Implements wear-leveling to prolong life of the flash media. Erase cycles are spread evenly across all erasable blocks. The wear count is maintained starting with the first time a flash volume is formatted. The current wear count, and other statistics, are available to applications via the `vstat()` call.
- Detects bad NAND blocks and performs bad block recovery without user intervention. After detection, bad blocks are neither written to nor erased.
- File system integrity is guaranteed across unexpected shutdowns. Only data written since the last synchronizing operation (`fclose()`, `fflush()`, etc.) can be lost. Closed files, directory structures, and files open for reading are never at risk.
- “Thin” driver layer for maximum ease in porting to new platforms. The driver performs only the basic operations that NAND devices support: read page, write page, erase block, etc. Sample drivers are provided for several chips and these are easily ported to new devices.
- Optimized seeks: `fseekmark()` and `lseekmark()` routines bookmark a file offset for faster seek performance, useful for no-glitch user interface requirements in MP3 players.
- Reduced footprint. Special-purpose memory manager reduces the amount of memory required for file names, enabling applications that have a small memory budget and require many files.
- Allows concurrent use of multiple volumes. Flash volumes can be permanently installed at startup, or added and deleted as removable devices come and go. Volume size is unlimited. A single volume can be implemented using multiple NAND devices.
- Supports the POSIX “self”, “group”, and “other” file access protections, allowing applications to restrict some operations to privileged tasks. The file system calls `FsGetId()`, implemented by the application, to get the running task’s user and group IDs.
- UTF-8 support allows non-ASCII file names, a key requirement for the Asian handset market. UTF-8 file name support can be enabled as a compile-time option.
- Includes fast and efficient error correction code (ECC) routines. Detects up to 8 bit errors and corrects up to 4 bit errors. Hardware based ECC can be used if available.
- Optimized for fast mounts. Typical mount time for a 64MB volume is approximately one second.
- Per-task current working directories. The current working directory (CWD) is specified by two 32-bit variables. TargetFFS-NAND calls two application functions to read and write the CWD state variables. If a real time operating system is used and these variables are accessed in a task specific way, each task has its own CWD.
- Garbage collection is performed to ensure minimal use of RAM for file system control information. Uses internal memory manager ▶

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

File Systems

High performance file systems for Flash, HD, and RAM. Implements power fail recovery, access protection, links, quotas, and other advanced features. Supports POSIX API.

TargetTCP

High Performance TCP/IP Stack with Standard Sockets API

TargetWeb

Embedded web server allows monitoring (data output) and control (data input) via desktop browser. Requires TargetTCP and any POSIX file system.

- for storing file names, reducing the RAM footprint when the average name length is much less than FILENAME_MAX. Recycle operations, which convert dirty sectors to free sectors, may be performed in the background by calling `vclean()` from a low priority task.
- Allows a number of flash sectors to be reserved, producing an early volume full indication. Free reserved sectors are immediately exchanged for dirty application sectors. When combined with background recycling, this boosts file system responsiveness for user interface applications, even when the volume is full or nearly full.
- Provides `fflush()` and `sync()` for application control of file system synchronization. Supports atomic file updates using `vclean()` to convert dirty sectors to free and `vstat()` to determine the number of available free sectors before the next recycle is required.
- Supports creation of special fixed length files using `creatn()` for reduced file system overhead and higher write performance.
- Shipped with five sample applications: a binary search application, a power-loss recovery test, a shell that supports “cd”, “ls”, “mkdir”, “pwd”, etc, and two applications that test assorted file system calls. The shell may be extended with user commands.
- Source code is 100% ANSI C and has been tested using ANSI C compilers from ARM, Diab Data, GNU, and Metrowerks.
- Developed using TargetOS™, Blunk Microsystems’ full-featured royalty-free real time operating system. Easily ported to other operating systems. Uses per-volume access semaphores to allow independent concurrent access to multiple volumes.
- Royalty-free site license. Includes source code, user’s manual, sample applications, and one year of technical support. ■

The TargetFFS-NAND Application Program Interface:

<code>access()</code>	<code>fopen()</code>	<code>isatty()</code>	<code>rewinddir()</code>
<code>chdir()</code>	<code>format()</code>	<code>link()</code>	<code>rmdir()</code>
<code>chmod()</code>	<code>fpathconf()</code>	<code>lseek()</code>	<code>scanf()</code>
<code>chown()</code>	<code>fprintf()</code>	<code>lseekmark()</code>	<code>setbuf()</code>
<code>clearerr()</code>	<code>fputc()</code>	<code>mkdir()</code>	<code>setvbuf()</code>
<code>close()</code>	<code>fputs()</code>	<code>mount()</code>	<code>sortdir()</code>
<code>closedir()</code>	<code>fread()</code>	<code>open()</code>	<code>stat()</code>
<code>creat()</code>	<code>freopen()</code>	<code>opendir()</code>	<code>sync()</code>
<code>dup()</code>	<code>fscanf()</code>	<code>pathconf()</code>	<code>tmpfile()</code>
<code>dup2()</code>	<code>fseek()</code>	<code>perror()</code>	<code>tmpnam()</code>
<code>fclose()</code>	<code>fsetpos()</code>	<code>printf()</code>	<code>truncate()</code>
<code>fcntl()</code>	<code>fstat()</code>	<code>putc()</code>	<code>ungetc()</code>
<code>fdopen()</code>	<code>ftell()</code>	<code>putchar()</code>	<code>unlink()</code>
<code>feof()</code>	<code>ftruncate()</code>	<code>puts()</code>	<code>unmount()</code>
<code>fflush()</code>	<code>fwrite()</code>	<code>read()</code>	<code>utime()</code>
<code>fgetc()</code>	<code>getc()</code>	<code>readdir()</code>	<code>vfprintf()</code>
<code>fgetpos()</code>	<code>getchar()</code>	<code>remove()</code>	<code>vsprintf()</code>
<code>fgets()</code>	<code>getcwd()</code>	<code>rename()</code>	<code>vstat()</code>
<code>fileno()</code>	<code>gets()</code>	<code>rewind()</code>	<code>write()</code>

Licensing Terms

TargetFFS-NAND™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

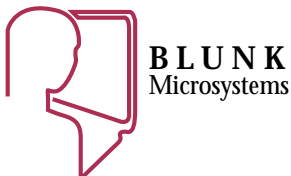
TargetFFS-NOR™

FLASH FILE SYSTEM

HIGHLIGHTS

- Reliable, Re-entrant Embedded File System
- Provides POSIX and Standard C APIs
- Guaranteed Integrity Across Unexpected Resets
- Use of NOR Flash Memory is Invisible to Applications
- Supports Dynamic Creation and Deletion of Files, Directories, and Links
- Performs Wear-Leveling
- Supports Multiple Volumes of Unlimited Size
- Licensed as Royalty-Free Source Code
- Portable to Virtually any RTOS and Toolchain
- Includes Sample Applications

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



TargetFFS-NOR is a reliable, re-entrant embedded file system with POSIX and ANSI C compliant application program interfaces (APIs). Supports dynamic creation and deletion of files, directories, and links with read and write capability. Not a static ROM-image file system. Use of NOR flash memory for the backing store is invisible to the application layer.

File system integrity is guaranteed across unexpected shutdowns. Only data written since the last synchronizing operation (`fclose()`, `fflush()`, etc.) can be lost. Directory structures, closed files, and files open for reading are never at risk.

Implements wear-leveling to prolong the life of the flash memory. Erase cycles are spread evenly across all erasable blocks. A wear count is maintained starting with the first time a flash volume is formatted and the current wear count is available to applications via the `vstat()` call.

Supports the large erasable block sizes typical of NOR flash memory, usually 64KB or larger. Memory is logically divided into 512 byte pages that are assigned to individual files as needed. Before a block is erased, the pages in use are copied to another block.

“Thin” driver layer performs only the basic operations that flash devices support: erase block, write byte, etc. The driver is stateless. Supports 8, 16, or 32-bit interfaces to the CPU. Wider interfaces and interleaved devices provide higher performance. Sample drivers for several NOR chips are provided and these are easily ported to new devices.

Supports concurrent use of multiple volumes. Flash volumes can be permanently installed at startup, or added and deleted as removable devices come and go. Volume size is unlimited. A single volume can be implemented using multiple NOR devices.

Flash memory can be shared between a boot program and a TargetFFS-NOR volume. Because flash memory is not accessible while being erased or programmed, the boot program can read files, change directories, etc., but cannot create files, directories, or links. Alternatively, the boot program can copy itself to RAM and jump there before making calls to TargetFFS-NOR.

Supports the UNIX “self”, “group”, and “other” file access protections, allowing applications to restrict some operations to privileged tasks. The file system calls `FsGetId()`, implemented by the application, to get the running task’s user and group IDs.

Optimized for fast mounts. Ongoing garbage collection ensures minimal use of RAM. Can coexist with TargetFFS-NAND™, Blunk Microsystems’ NAND flash file system.

Sample applications include a binary search test, a boot loader, and a command line shell. The boot loader recovers from power loss during an application update. It loads the most recent copy that has been successfully written to flash. The shell supports “cd”, “ls”, “mkdir”, “pwd”, etc. and may be extended with user commands.

Developed using TargetOS™, Blunk Microsystems’ real-time operating system, TargetFFS-NOR has been ported to several other real-time kernels including Nucleus, VxWorks, and Precise MQX. Can be used without a kernel. Works on big- and little-endian systems.

The TargetFFS-NOR source code is 100% ANSI C and is integrated with CodeWarrior. It has been used with ANSI C compilers from ARM, Diab Data, GNU, Software Development Systems, and WindRiver.

Royalty free. Includes full source code, user’s manual, sample applications, and one year of technical support. ►

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

OTHER COMPONENTS

TargetOS

Scalable, Deterministic, Priority-Based Preemptive Real-Time Kernel

TargetTCP

High Performance TCP/IP Stack with Standard Sockets API

TargetLAPB

LAPB Stack for Fast Data Transfer on Point-to-Point Networks

TargetWeb

HTTP 1.0 and 1.1 Compliant Embedded Web Server

TargetZFS

Compressed Data Read-Only Embedded File System

TargetFFS-NAND

Reliable Power-Fail-Safe Embedded NAND Flash File System

The TargetFFS-NOR Application Program Interface:

```
int access(const char *path, int amode);
int chdir(const char *path);
int chmod(const char *path, mode_t mode);
int chown(const char *path, uid_t owner, gid_t group);
void clearerr(FILE* stream);
int close(int fid);
int closedir(DIR *dirp);
int dup(int fid);
int dup2(int fid, int fid2);
int fclose(FILE *stream);
int fcntl(int fid, int cmd, ...);
FILE *fdopen(int fid, const char *mode);
int feof(FILE *stream);
int ferror(FILE *stream);
int fflush(FILE *stream);
int fgetc(FILE *stream);
int fgetpos(FILE *stream, fpos_t *pos);
char *fgets(char *s, int n, FILE *stream);
int fileno(FILE *stream);
FILE *fopen(const char *filename, const char *mode);
int format(char *path);
int fprintf(FILE *stream, const char *format, ...);
int fputc(int c, FILE *stream);
int fputs(const char *string, FILE *stream);
size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);
FILE *freopen(const char *filename, const char *mode, FILE *stream);
int fscanf(FILE *stream, const char *format, ...);
int fseek(FILE *stream, long offset, int mode);
int fsetpos(FILE *stream, const fpos_t *pos);
int fstat(int fid, struct stat *buf);
long ftell(FILE *stream);
int ftruncate(int fid, off_t length);
size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);
int getc(FILE *stream);
int getchar(void);
char *getcwd(char *buf, size_t size);
char *gets(char *s);
int isatty(int fid);
int link(const char *existing, const char *new);
off_t lseek(int fid, off_t offset, int whence);
int mkdir(const char *path, mode_t mode);
int mount(char *path);
int open(const char *path, int oflag, ...);
DIR *opendir(const char *dirname);
void perror(const char *s);
int printf(const char *format, ...);
int putc(int c, FILE *stream);
int putchar(int c);
int puts(const char *s);
int read(int fid, void *buf, unsigned int nbyte);
struct dirent *readdir(DIR *dirp);
int remove(const char *filename);
int rename(const char *old, const char *new);
void rewind(FILE *stream);
void rewinddir(DIR *dirp);
int rmdir(const char *path);
int scanf(const char *format, ...);
void setbuf(FILE *stream, char *buf);
int setvbuf(FILE *stream, char *buf, int mode, size_t size);
int stat(const char *path, struct stat *buf);
int sync(int fid);
FILE *tmpfile(void);
char *tmpnam(char *s);
int truncate(const char *path, off_t length);
int ungetc(int c, FILE *stream);
int unlink(const char *path);
int unmount(char *path);
int utime(const char *path, const struct utimbuf *times);
int vfprintf(FILE *stream, const char *format, va_list arg);
int vprintf(const char *format, va_list arg);
int vstat(const char *path, union vstat *buf);
int write(int fid, const void *buf, unsigned int nbyte); ■
```

Licensing Terms

TargetFFS-NOR™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

EMBEDDED WEB SERVER

HIGHLIGHTS

- Monitor and control embedded applications using any browser from anywhere
- Server replaces custom <DATA> tags on the fly with user function supplied content
- Dynamic handlers (custom functions) for generating HTML content as required by forms
- Update web pages via FTP
- Uses any POSIX compatible file system
- Accessible via Ethernet, serial, Wi-Fi, etc.
- Integrated with TargetOsTools™

Blunk Microsystems provides development tools, system software, device drivers, and board support packages to the embedded systems market.



Compliant with HTTP protocol versions 1.0 and 1.1.

Allows monitoring and control of embedded applications via any standard browser. If the device designer sets up the mechanism, supports execution of target code, thereby enabling access to virtually any operational functionality of the target.

Custom <DATA> tags embedded in HTML pages allow the server to provide updateable data on the fly as specified by the user.

Simple mechanism for generating dynamic HTML, such as that required by the use of Forms.

Tightly integrated with TargetTCP™, Blunk Microsystems' embedded TCP/IP protocol stack. Uses TargetTCP's zero-copy and event callback interfaces for high performance.

Easy to configure and use. The user specifies (1) the server's home directory and (2) the file containing the home page. The HTML pages can be updated using TargetTCP's built-in FTP server.

TargetWeb uses POSIX API to access the underlying file system. Use with TargetFFS™, Blunk Microsystems' industry-leading embedded flash file system, allows the web pages to be dynamically updated, not hard-coded, and allows operation in environments where temperature, shock, or power restrictions prohibit use of a hard disk. Use with TargetZFS™ to serve compressed pages or to link a file system in with your application code.

Alternatively, it can be used in conjunction with disk based or any other available file system.

Compact. Adds approximately 30 KB of code and 30 KB of data to the application.

Shipped with a sample application including sample web content that can be used as the starting point for your application.

Source code is 100% ANSI C and has been tested using the gcc and CodeWarrior ANSI C compilers.

Developed using TargetOS™, Blunk Microsystems' real-time operating system. TargetTCP, TargetWeb, and the flash file systems are easily ported to other real-time kernels or to polling environments that do not use a kernel.

Royalty-free. Includes source code, user's manual, sample applications, and one year of technical support. ■

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

RFC compliant TCP/IP protocol stack providing the standard Berkeley Sockets API.

File Systems

High performance file systems for Flash, HD, and RAM. Implements power fail recovery, access protection, links, quotas, and other advanced features. Supports POSIX API.

Licensing Terms

TargetWeb™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

TargetFAT™

DOS/Win-Compatible File System

HIGHLIGHTS

- POSIX and ANSI C compliant API
- FAT12, FAT16 and FAT32 formats
- VFAT and UTF8 support
- Works with Hard Drives and CompactFlash cards
- Works with flash translation layers (TargetFTL) for NAND and NOR memory
- Uses the same API files and caching routines as TargetFFS™ (NAND and NOR flash memory), TargetRFS™ (RAM), and TargetZFS™ (compressed read-only)

Blunk Microsystems provides development tools, system software, device drivers, and board support packages to the embedded systems market.



TargetFAT™, Blunk Microsystems' DOS/Win-Compatible File System, includes the following features:

A reliable, re-entrant file system with a POSIX and ANSI C compliant application program interface.

Supports the DOS/Windows FAT12, FAT16, and FAT32 formats.

Long file name support (VFAT) and UTF8 file name support can be enabled as compile-time options.

Within the ranges allowed by the FAT12, FAT16, and FAT32 formats, allows specification of the cluster size.

Works with hard drives and CompactFlash cards. Supported by TargetFTL-NAND™ and TargetFTL-NOR™, Blunk Microsystems' flash translation layers for NAND and NOR memory.

Uses the same API files and caching routines as Blunk Microsystems other file systems: TargetFFS™ (NAND and NOR flash memory), TargetRFS™ (RAM), and TargetZFS™ (compressed read-only). Works seamlessly with these other systems.

Shipped with four sample applications: a binary search application, a shell that supports "cd", "ls", "mkdir", "pwd", etc, and two applications that test assorted file system calls. The shell may be extended with user commands.

Supports multiple volumes.

Integrated with TargetOS™, Blunk Microsystems' real-time operating system. The file system is easily ported to other real-time kernels or to polling environments that do not use a kernel.

Royalty-free Site License. Includes ANSI C source code, user's manual, sample applications, and one year of technical support. ■

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

File Systems

High performance file systems for Flash, HD, and RAM. Implements power fail recovery, access protection, links, quotas, and other advanced features. Supports POSIX API.

TargetTCP

High Performance TCP/IP Stack with Standard Sockets API

TargetWeb

Embedded web server allows monitoring (data output) and control (data input) via desktop browser. Requires TargetTCP and any POSIX file system.

The TargetFAT™ Application Program Interface:

<code>access()</code>	<code>fopen()</code>	<code>isatty()</code>	<code>rewinddir()</code>
<code>chdir()</code>	<code>format()</code>	<code>link()</code>	<code>rmdir()</code>
<code>chmod()</code>	<code>fpathconf()</code>	<code>lseek()</code>	<code>scanf()</code>
<code>chown()</code>	<code>fprintf()</code>	<code>lseekmark()</code>	<code>setbuf()</code>
<code>clearerr()</code>	<code>fputc()</code>	<code>mkdir()</code>	<code>setvbuf()</code>
<code>close()</code>	<code>fputs()</code>	<code>mount()</code>	<code>sortdir()</code>
<code>closedir()</code>	<code>fread()</code>	<code>open()</code>	<code>stat()</code>
<code>creat()</code>	<code>freopen()</code>	<code>opendir()</code>	<code>sync()</code>
<code>dup()</code>	<code>fscanf()</code>	<code>pathconf()</code>	<code>tmpfile()</code>
<code>dup2()</code>	<code>fseek()</code>	<code>perror()</code>	<code>tmpnam()</code>
<code>fclose()</code>	<code>fsetpos()</code>	<code>printf()</code>	<code>truncate()</code>
<code>fcntl()</code>	<code>fstat()</code>	<code>putc()</code>	<code>ungetc()</code>
<code>fdopen()</code>	<code>ftell()</code>	<code>putchar()</code>	<code>unlink()</code>
<code>feof()</code>	<code>ftruncate()</code>	<code>puts()</code>	<code>unmount()</code>
<code>ferror()</code>	<code>fwrite()</code>	<code>read()</code>	<code>utime()</code>
<code>fflush()</code>	<code>getc()</code>	<code>readdir()</code>	<code>vfprintf()</code>
<code>fgetc()</code>	<code>getchar()</code>	<code>remove()</code>	<code>vsprintf()</code>
<code>fgetpos()</code>	<code>getcwd()</code>	<code>rename()</code>	<code>vstat()</code>
<code>fgets()</code>	<code>gets()</code>	<code>rewind()</code>	<code>write()</code>
<code>fileno()</code>			

Licensing Terms

TargetFAT™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

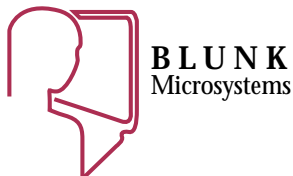
TargetLAPB™

LAPB PROTOCOL STACK

HIGHLIGHTS

- Royalty Free
- ISO/IEC 7776 Compliant
- Supports Point-to-Point Networks
- Provides Flow Control and Reliability
- Configurable T1, T2, T4, N2, Window Size, Sequence Numbering, and Addressing
- Supports Multiple HDLC Channels
- Built-in LAPB Protocol Decoder
- Works with TargetOS
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



TargetLAPB is compliant with International Standard ISO/IEC 7776:1995(E) "Description of the X.25 LAPB-compatible DTE data link procedures" and has been extensively tested and demonstrated interoperable with other LAPB protocol stacks.

TargetLAPB supports exchanging data on point-to-point networks and provides automatic flow control and data reliability. Reliability is provided using CRC checks, frame sequence numbering, and positive acknowledgments with resends after timeouts. Flow control is provided using LAPB's sliding window protocol.

Configuration parameters T1, T2, T4, N2, window size, sequence number modulus (standard or extended), and address (DTE or DCE) are independently programmable for each channel.

LAPB events are reported to the application layer using an optional callback function. The following events are reported: N2 unacknowledged SABM/SABME commands, N2 unacknowledged DISC commands, N2 RNR responses, frame reject, physical disconnect, link reset, link reconnect, all I-frames acknowledged, and I-frame arrival.

Statistics are maintained on the number of connections/reconnections, number of frames sent and received, etc. The statistics may be read and/or written to `stdout` with the `lapbStatistics()` call.

Built-in LAPB protocol decoder. When enabled by a compile-time flag, a short summary of every packet transmitted or received is printed to `stdout` on an ongoing basis as the protocol stack runs.

TargetLAPB has a clearly documented HDLC driver interface and supports concurrent use of multiple channels and multiple serial controllers. HDLC drivers for Motorola MPC8XX, MC68360, and MC68302 SCC channels and other serial modules are available separately. Blunk Microsystems provides competitive bids on custom drivers.

To help catch errors during development, a compile-time option promotes TargetLAPB service call parameter errors to fatal errors that break into the command line monitor after printing an error message to `stderr`.

Shipped with a sample application designed to endlessly send I-frames to a LAPB test set operating in "echo" mode. All transmitted data is echoed back to the application, which tests the returning data for errors. The sample application is a good test of the TargetLAPB configuration, HDLC driver integration, and physical cabling.

Developed using TargetOS™, Blunk Microsystems' real time operating system, the source code is 100% ANSI C and is easily ported to other real-time kernels.

Integrated with CodeWarrior™, the development environment from Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool.

Royalty free. Includes full source code, user's manual, sample application, and one year of technical support. ►

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

RFC compliant TCP/IP protocol stack providing the standard Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system. Implements wear-leveling to prolong life of flash media. File system integrity is guaranteed across unexpected resets.

The TargetLAPB Application Program Interface:

```
int lapbInit(void (*report)(int chan, ui32 event), void (*parse)(ui8 *data));  
Initializes TargetLAPB and provides two optional call-back functions.
```

```
int lapbConfigure(int channel, LapbSpec *lapb_spec);  
Configures a channel. LapbSpec defines the LAPB configuration.
```

```
typedef struct  
{  
    uint T1;           /* resend timeout (tenths of seconds) */  
    uint T2;           /* acknowledgment delay (tenths of seconds) */  
    uint T4;           /* idle channel timeout (tenths of seconds) */  
    uint N2;           /* max retry value */  
    uint window_size; /* max window size (less than 8/128) */  
    ui8 seq_modulus;   /* LAPB_SEQ_MOD_8 or LAPB_SEQ_MOD_128 */  
    ui8 addr_type;     /* LAPB_DTE_ADDR or LAPB_DCE_ADDR */  
} LapbSpec;
```

```
int lapbConnect(int channel, int send_sabm);  
Establishes a channel's LAPB connection.
```

```
int lapbDisconnect(int channel, uint flags);  
Terminates a channel's LAPB and/or physical connection.
```

```
int lapbSend(int channel, void *buffer, int len);  
Transmits an application buffer as the data field in a LAPB I-frame.
```

```
int lapbRecv(int channel, void *buffer);  
Receives data from a LAPB channel.
```

```
int lapbStatistics(int channel, int print, LapbStats *lp, PhysStats *pp);  
Accesses a channel's statistics. If print is TRUE, a report is written to stdout. ■
```

Licensing Terms

TargetLAPB™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

TargetOS Board Support Package

HIGHLIGHTS

- Hosts TargetOS on phyCORE®-MCF5485
- 100 Mbps Ethernet Driver
- UART and Kernel Tick Drivers
- Stores Configuration Parameters in Flash
- Flash File System Driver
- MCF5485 MMU, Data and Instruction Cache Support
- Integrated with TargetOsTools™

Blunk Microsystems provides development tools, system software, device drivers, and board support packages to the embedded systems market.



Blunk Microsystems' board support package for phyCORE®-MCF5485, PHYTEC's single board computer based on Freescale's MCF5485 ColdFire processor, includes the following features:

Performs low level initialization of the MCF5485 including CPU, Communications I/O Subsystem, System Integration Unit (SIU) and SDRAM controller, allowing TargetOS™ applications to boot from external Flash.

Drivers for the MCF5485's dual Fast Ethernet Controllers (FECs). The driver uses the highest negotiable combination of half and full duplex, 10 and 100 Mbps, and implements a zero-copy interface to TargetTCP™, Blunk Microsystems' high performance TCP/IP stack. Minimal time is spent in the FEC interrupt handler. Supports TFTP downloads, FTP transfers, Telnet access to the TargetOS command line monitor, and other TCP/IP applications.

Interrupt-driven and polled-mode drivers for MCF5485 Programmable Serial Controller (PSC) channels PSC0 and PSC1 in UART mode. Supports full-duplex operation of the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C library. The baud rate is set independently for each channel. Default configuration is 8-bit data, 1 stop bit, 1 start bit, and no parity.

Driver for MCF5485 Slice Timer 1 supports the TargetOS 32-bit system timer and the kernel tick interrupt used for task sleeps and system call timeouts. The default tick interrupt frequency is 100 Hz.

Drivers for MCF5485 General Purpose Timers 0-3 allow either periodic or one-time calls to user-provided callback functions after programmed delays. The default resolution is 1 us and the maximum delay is 71.6 min.

Supports the MCF5485 MMU. Accesses to unsupported address ranges are caught and reported to the application as bus errors.

Supports the MCF5485 data and instruction caches. The instruction cache is enabled for all accesses. Data accesses to the SDRAM use copy-back cache mode with imprecise ordering. Flash data accesses are cache inhibited with imprecise ordering. All other data accesses are cache-inhibited with precise ordering. Provides service calls to flush, invalidate, and synchronize the instruction and data caches.

A menu-driven Flash programmer for installing TargetOS boot applications into the external flash. The programmer supports the ELF and S-record formats and accepts input from either a file, UART channel, or TFTP connection. The executable image is written to flash after it is successfully loaded to a RAM buffer. ▶

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multitasking, priority-based, preemptive kernel. Includes Standard C library.

File Systems

High performance file systems for Flash, HD, and RAM. Implements power fail recovery, access protection, links, quotas, and other advanced features. Supports POSIX API.

TargetTCP

High Performance TCP/IP Stack with Standard Sockets API.

TargetOsTools

Embedded toolset for Windows. Includes gcc and Insight/gdb port, TargetOS, advanced kernel awareness with kernel event trace display, an Ethernet-capable debug monitor for fast downloads, BDM debug support, and GUI build tool with automatic time-stamp checking.

Uses one block of flash memory to support the TargetOS NVRAM driver interface, providing non-volatile configuration parameters. These parameters, such as startup mode, baud rate, and network configuration, can be displayed and edited via the TargetOS boot menu. Applications can add their own configuration parameters and boot menu entries.

Integrated with TargetOsTools™/ColdFire, the IDE for embedded development from Blunk Microsystems with an integrated compiler and kernel-aware debugger, project manager, visual code editor, search and replace tool, BDM for board bring-up, and fast Ethernet download.

Royalty-free. Site licensed, allowing use on unlimited number of projects. Includes full source code, user's manual, and one year of technical support. ■

Licensing Terms

phyCORE®-MCF5485 BSP is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

M5282EVB BSP

TargetOS Board Support Package

HIGHLIGHTS

- Hosts TargetOS on M5282EVB
- 100 Mbps Ethernet Driver
- UART and Kernel Tick Drivers
- Stores Configuration Parameters in Flash
- Flash File System Driver
- Updates Boot Program via TFTP
- Low Level Initialization Code
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



Provides initialization routines for hosting TargetOS, Blunk Microsystems' high performance real time operating system, on Motorola's M5282EVB evaluation board. Performs low-level initialization of the CPU, System Integration Module, and SDRAM controller, allowing TargetOS applications to boot from external Flash.

Includes driver for the MCF5282's Fast Ethernet Controller (FEC). The driver uses the highest negotiable combination of half and full duplex, 10 and 100 Mbps, and implements a zero-copy interface to TargetTCP™, Blunk Microsystems' high performance TCP/IP stack. Minimal time is spent in the FEC interrupt handler. Supports TFTP downloads, FTP transfers, Telnet access to the TargetOS command line monitor, and other TCP/IP applications.

Interrupt-driven and polled-mode drivers for MCF5282 UART channels 0 and 1. Supports full-duplex operation of the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C library. The baud rate is set independently for each channel. Default configuration is 8-bit data, 1 stop bit, 1 start bit, and no parity.

Driver for MCF5282 DMA timer 3 supports the TargetOS 32-bit system timer and the kernel tick interrupt used for task sleeps and system call timeouts. The default tick interrupt frequency is 100 Hz.

Drivers for MCF5282 DMA timers 0-2 allow either periodic or one-time calls to user-provided callback functions after programmed delays. The default resolution is 1 μ s and the maximum delay is 71.6 minutes.

Implements service calls to flush, invalidate, and synchronize the instruction and data caches.

Includes a menu-driven Flash programmer for installing TargetOS boot applications into the external flash. The programmer supports the ELF and S-record formats and accepts input from a file, UART channel, or TFTP connection. The executable image is written to flash after it is successfully loaded to a RAM buffer.

Uses one block of flash memory to support the TargetOS NVRAM driver interface, providing non-volatile configuration parameters. These parameters, such as startup mode, baud rate, and network configuration, can be displayed and edited via the TargetOS boot menu. Applications can add their own configuration parameters and boot menu entries.

Integrated with CodeWarrior™, the development environment from Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool. Includes project build file, JTAG initialization file, and linker command file.

Royalty-free. Site licensed, allowing use on unlimited number of projects. Includes full source code, user's manual, and one year of technical support. ■

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

TCP/IP protocol stack with Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system with wear-leveling and integrity across unexpected resets.

TargetLAPB

ISO/IEC 7776 protocol stack provides flow control and reliability on point-to-point networks.

Licensing Terms

Blunk Microsystems' M5282EVP BSP is royalty free. Purchasers are granted a non-exclusive license to use, modify, or extract the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

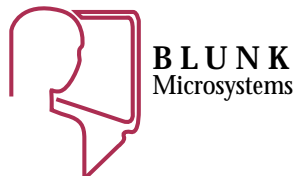
M5272C3 BSP

TargetOS Board Support Package

HIGHLIGHTS

- Hosts TargetOS on M5272C3
- UART and Tick Interrupt Drivers
- Fast Ethernet Driver
- Flash Memory Programmer
- Low Level Initialization Code
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



Drivers and initialization routines for hosting TargetOS™, Blunk Microsystems' high performance real-time operating system, on Motorola's M5272C3 evaluation board.

Ethernet driver for the MCF5272's 100 Mbps FEC supports TargetTCP™, Blunk Microsystems' high performance TCP/IP stack. The driver allows TFTP downloads, FTP transfers, and other network applications.

Interrupt-driven and polled-mode driver for the MCF5272's UART Modules supports the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C library. Baud rate, parity, flow control mode, and number of stop bits are programmable.

Periodic tick interrupt driver supports the task sleeps and service call timeouts provided by TargetOS. The default tick interrupt frequency is 100 times per second.

The MCF5272 instruction cache is enabled and a service call (`SyncCache()`) is provided for flushing the cache whenever new code has been loaded into RAM or Flash.

Includes low level initialization code for the CPU, System Integration Module, and SDRAM controller, allowing TargetOS applications to boot from Flash.

Contains routines for managing configuration parameters such as startup mode, baud rate, and IP address. These parameters are assigned default values and can be modified using the TargetOS boot menu.

Driver for MCF5272 general-purpose timer module allows applications to generate one-shot and periodic interrupts with a maximum period of 4 seconds and a maximum resolution of 15 nanoseconds (at 66 MHz). User callback functions are installed as the timer interrupt handlers.

Flash programmer allows boot program updates via the TargetOS boot menu. Upon request, an S-record file is downloaded over the serial port and installed in the M5272C3's 16-bit boot flash.

Integrated with CodeWarrior™, the development environment from Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool. Includes kernel awareness DLL.

Royalty free. Includes full ANSI C source code, default compiler settings, linker command files, user's manual, and one year of technical support. ■

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

TCP/IP protocol stack with Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system with wear-leveling and integrity across unexpected resets.

TargetLAPB

ISO/IEC 7776 protocol stack provides flow control and reliability on point-to-point networks.

Licensing Terms

Blunk Microsystems' M5272C3 BSP is royalty free. Purchasers are granted a non-exclusive license to use, modify, or extract the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

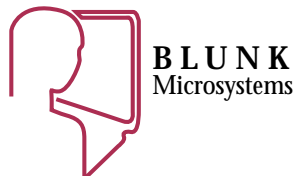
MPC8XX FADS BSP

TargetOS Board Support Package

HIGHLIGHTS

- Hosts TargetOS on MPC8XX FADS
- UART and Tick Interrupt Drivers
- Supports Flash File System
- Time-of-Day Clock Driver
- 100/10 Mbps Ethernet Drivers
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



Provides drivers and initialization routines for hosting TargetOS™, Blunk Microsystems' high performance real-time operating system, on the MPC8XX FADS, Motorola's MPC8XX evaluation board.

Interrupt-driven and polled-mode UART drivers for the MPC8XX SMC1 channel supports the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C library. Baud rate, parity, flow control mode, and number of stop bits are programmable.

Periodic tick interrupt driver supports the task sleeps and service call timeouts provided by TargetOS. The default tick interrupt frequency is 100 times per second.

Enables the MPC8XX instruction and data caches for maximum performance. The DMMU is enabled to support speculative accesses, a performance boost, and to support address-selective data caching, required in order to not cache I/O registers.

Time-of-Day clock driver for the MPC8XX's real-time clock supports the time-related functions in the TargetOS Standard C library. The time-of-day and date can be set by application programs or by the TargetOS boot menu.

MPC8XX general-purpose timer driver allows applications to generate one-shot and periodic interrupts with microsecond resolution. User callback functions are installed as the timer interrupt handlers.

Ethernet drivers for the MPC8XX's 100 Mbps FEC and 10 Mbps SCC1 support TargetTCP™, Blunk Microsystems' high performance TCP/IP stack, allowing FTP transfers, Telnet sessions, TFTP downloads, and other network applications.

Flash programmer allows boot program updates via the TargetOS boot menu. Upon request, an S-record file is downloaded over the serial port and installed as a boot program in the FADS flash memory.

Driver for TargetFFS™, Blunk Microsystems' high performance flash file system, allows using a portion of the FADS flash memory as a file system volume. TargetFFS has a POSIX and Standard C compliant API and provides guaranteed file system integrity across unexpected shutdowns.

Integrated with CodeWarrior™, the development environment from Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool. Includes kernel awareness DLL.

Royalty free. Includes full ANSI C source code, default compiler settings, linker command files, user's manual, and one year of technical support. ■

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

TCP/IP protocol stack with Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system with wear-leveling and integrity across unexpected resets.

TargetLAPB

ISO/IEC 7776 protocol stack provides flow control and reliability on point-to-point networks.

Licensing Terms

Blunk Microsystems' MPC8XX FADS BSP is royalty free. Purchasers are granted a non-exclusive license to use, modify, or extract the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

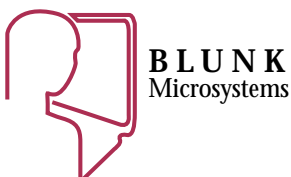
XP860T BSP

TargetOS Board Support Package

HIGHLIGHTS

- Hosts TargetOS on XP860T
- UART and Tick Interrupt Drivers
- Stores Configuration Parameters in NVRAM
- Supports Flash File System
- Time-of-Day Clock Driver
- 100/10 Mbps Ethernet Drivers
- Synchronous HDLC Driver
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



Provides drivers and initialization routines for hosting TargetOS™, Blunk Microsystems' high performance real-time operating system, on the XP860T, JRM Consultant's single board computer.

Interrupt-driven and polled-mode UART drivers for the MPC860 SMC1 channel supports the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C library. Baud rate, parity, flow control mode, and number of stop bits are programmable.

Periodic tick interrupt driver supports the task sleeps and service call timeouts provided by TargetOS. The default tick interrupt frequency is 100 times per second.

For maximum performance, enables the MPC860 instruction and data caches. The DMMU is enabled in order to support speculative accesses, a performance boost, and to support address-selective data caching, required in order to not cache I/O registers.

Time-of-Day clock driver for the XP860T's real-time clock supports the time-related functions in the TargetOS Standard C library. The time-of-day and date can be set by application programs or by the TargetOS boot menu.

NVRAM driver stores configuration parameters such as startup mode, baud rate, and IP address in the XP860T's non-volatile memory. Configuration parameters have user-defined default values and can be modified using the TargetOS boot menu.

MPC860 general-purpose timer driver allows applications to generate one-shot and periodic interrupts with microsecond resolution. User callback functions are installed as the timer interrupt handlers.

Ethernet drivers for the XP860T's 100 Mbps FEC and 10 Mbps SCC1 support TargetTCP™, Blunk Microsystems' high performance TCP/IP stack. The drivers allow TFTP downloads, FTP transfers, and other network applications.

Flash programmer allows boot program updates via the TargetOS boot menu. Upon request, an S-record file is downloaded over the serial port and installed in the XP860T's boot flash.

Flash driver for the XP860T's flash file memory supports TargetFFS™, Blunk Microsystems' high performance flash file system. TargetFFS provides a POSIX and Standard C compliant API and provides guaranteed file system integrity across unexpected shutdowns.

Synchronous HDLC driver for the XP860T's SCC2 multi-protocol interface supports TargetLAPB™, Blunk Microsystems' high performance ISO/IEC 7776 compliant LAPB protocol stack for exchanging data on point-to-point

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

TCP/IP protocol stack with Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system with wear-leveling and integrity across unexpected resets.

TargetLAPB

ISO/IEC 7776 protocol stack provides flow control and reliability on point-to-point networks.

connections. Custom drivers for other synchronous protocols are available from Blunk Microsystems.

Integrated with CodeWarrior™, the development environment from Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool. Includes kernel awareness DLL.

Royalty free. Includes full ANSI C source code, default compiler settings, linker command files, user's manual, and one year of technical support. ■

Licensing Terms

Blunk Microsystems' XP860T BSP is royalty free. Purchasers are granted a non-exclusive license to use, modify, or extract the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

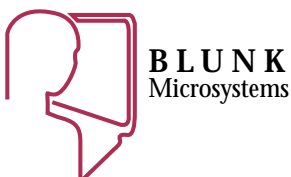
MBX860 BSP

TargetOS Board Support Package

HIGHLIGHTS

- Hosts TargetOS on MBX860
- UART and Tick Interrupt Drivers
- Stores Configuration Parameters in NVRAM
- Supports Flash File System
- Time-of-Day Clock Driver
- Ethernet Driver
- Synchronous HDLC Driver
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



Provides drivers and initialization routines for hosting TargetOS™, Blunk Microsystems' full-featured royalty free real-time operating system, on Motorola's MBX860 single board computer.

Interrupt-driven and polled-mode UART driver for the MPC860's SMC1 channel supports the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C library. Baud rate, parity, flow control mode, and number of stop bits are programmable.

Periodic tick interrupt driver supports the task sleeps and service call timeouts provided by TargetOS. The default tick interrupt frequency is 100 times a second.

Time-of-Day clock driver supports the time-related functions in the TargetOS Standard C library. The time-of-day and date can be set by application programs or by the TargetOS boot menu.

NVRAM driver stores configuration parameters such as startup mode, baud rate, and IP address in the MBX860's battery-backed SRAM. Configuration parameters are assigned default values and can be modified using the TargetOS boot menu.

MPC860 general-purpose timer driver allows applications to generate one-shot and periodic interrupts with microsecond resolution. User callback functions are installed as the timer interrupt handlers.

Ethernet driver for the MPC860's SCC1 channel supports TargetTCP™, Blunk Microsystems' royalty free TCP/IP stack and allows networked applications.

Flash programmer allows boot program updates as an option in the TargetOS boot menu. Upon request, an S-record file is downloaded over the serial port and installed in the MBX860's 8-bit boot flash.

Flash file system driver for the MBX860's 4/8 MBs of 32-bit flash memory supports TargetFFS™, Blunk Microsystems' royalty free flash file system. TargetFFS provides a POSIX and Standard C compliant API and provides guaranteed file system integrity across unexpected shutdowns.

SCC drivers (HDLC, UART, BISYNC, etc.) are available for the MBX860's COMM interface, including HDLC driver support for TargetLAPB™, Blunk Microsystems' royalty free ISO/IEC 7776 compliant LAPB protocol stack. TargetLAPB provides a low-overhead protocol for reliably exchanging data on a point-to-point connection.

Performs initialization and configuration of the MBX860's PCI bus.

Includes high performance DMA-mode driver for the MBX860's IDE interface.

Integrated with CodeWarrior™, the development environment from

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

TCP/IP protocol stack with Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system with wear-leveling and integrity across unexpected resets.

TargetLAPB

ISO/IEC 7776 protocol stack provides flow control and reliability on point-to-point networks.

Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool.

Royalty free. Includes full ANSI C source code, default compiler settings, linker command files, user's manual, and one year of technical support. ■

Licensing Terms

Blunk Microsystems' MBX860 BSP is royalty free. Purchasers are granted a non-exclusive license to use, modify, or extract the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

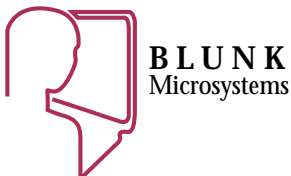
VIPC860 BSP

TargetOS Board Support Package

HIGHLIGHTS

- Hosts TargetOS on VIPC860
- UART and Tick Interrupt Drivers
- Stores Configuration Parameters in NVRAM
- Supports Flash File System
- Time-of-Day Clock Driver
- Fast Ethernet Driver
- Synchronous HDLC Driver
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



Provides drivers and initialization routines for hosting TargetOS™, Blunk Microsystems' full-featured royalty free real-time operating system, on the VIPC860, SBS Modular I/O's 6U VMEbus IP carrier.

Interrupt-driven and polled-mode UART drivers for both MPC860 SMC channels support the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C library. Baud rate, parity, flow control mode, and number of stop bits are programmable.

Periodic tick interrupt driver supports the programmable task sleeps and service call timeouts provided by TargetOS. The default tick interrupt frequency is 100 times a second.

Time-of-Day clock driver supports the time-related functions in the Standard C library. The time-of-day and date can be set from application programs or by using the TargetOS boot menu.

EEPROM driver stores configuration parameters such as startup mode, baud rate, and IP address in the VIPC860's non-volatile memory. Configuration parameters have user-defined default values and can be modified using the TargetOS boot menu.

Driver for the MPC860's general-purpose timers allows applications to generate either one-shot or periodic interrupts with microsecond resolution. C callback functions are installed as the timer interrupt handlers.

Ethernet driver for the MPC860 Fast Ethernet controller supports TargetTCP™, Blunk Microsystems' royalty free TCP/IP stack, and allows TFTP downloads, network connections to a source level debugger, and other TCP/IP applications.

Flash drivers install VIPC860 boot programs and support TargetFFS™, Blunk Microsystems' royalty free POSIX and Standard C compliant flash file system. The VIPC860's 4MB of flash memory can be shared between boot programs and flash file system volumes.

Synchronous HDLC driver for the MPC860 SCC channels supports TargetLAPB™, Blunk Microsystems' royalty free ISO/IEC 7776 compliant LAPB protocol stack for exchanging data on point-to-point connections. Custom drivers for other synchronous protocols are available from Blunk Microsystems.

Integrated with CodeWarrior™, the development environment from Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool.

Royalty free. Includes full ANSI C source code, default compiler settings, linker command files, user's manual, and one year of technical support. ►

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

TCP/IP protocol stack with Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system with wear-leveling and integrity across unexpected resets.

TargetLAPB

ISO/IEC 7776 protocol stack provides flow control and reliability on point-to-point networks.

The following IndustryPack drivers are available with the VIPC860 BSP:

Comm302 HDLC - TargetLAPB HDLC Driver

An HDLC driver for TargetLAPB. May also be used as a stand-alone HDLC frame driver or with other third party protocol stacks. Supports all three 68302 SCC channels.

Comm360 HDLC - TargetLAPB HDLC Driver

An HDLC driver for TargetLAPB. May also be used as a stand-alone HDLC frame driver or with other third party protocol stacks. Supports all four 68360 SCC channels.

IP Digital - IndustryPack Digital I/O Drivers

A family of drivers for IndustryPack digital I/O modules providing turnkey operation when used with TargetOS.

IP Ethernet - TargetTCP Ethernet Driver

An Ethernet driver for TargetTCP. Supports 10BASE-2, 10BASE-5, and 10BASE-T media interfaces.

IP UART - IndustryPack UART Drivers

A family of drivers for IndustryPack UART modules. Supports the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C Library. The drivers operate in both polled and interrupt driven mode. Configurable for either software or hardware flow control. ■

Licensing Terms

Blunk Microsystems' VIPC860 BSP is royalty free. Purchasers are granted a non-exclusive license to use, modify, or extract the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.

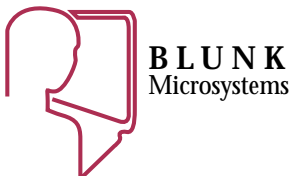
A3000 BSP

TargetOS Board Support Package

HIGHLIGHTS

- Hosts TargetOS on 400 MHz MPC8245
- 100 Mbps Ethernet Driver
- PCI Bus Scan and Initialization
- UART and Kernel Tick Drivers
- Stores Configuration Parameters in Flash
- Supports Flash File System
- Integrated with CodeWarrior

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.



Provides drivers and initialization routines for hosting TargetOS, Blunk Microsystems' high performance real-time operating system, on the A3000, ARTiS Microsystems' MPC8245-based single board computer.

Interrupt-driven and polled-mode driver for the MPC8245 DUART supports the stream I/O routines (`printf()`, `scanf()`, etc.) in the TargetOS Standard C library. Baud rate, parity, flow control mode, and number of stop bits are programmable.

Ethernet driver for the A3000's DP83815 Fast Ethernet controller. The driver automatically detects and adapts to the highest negotiable combination of half and full duplex, 10 and 100 Mbps, and uses the zero copy interface to TargetTCP™, Blunk Microsystems' high performance TCP/IP stack. This allows TFTP downloads, FTP transfers, Telnet access to the TargetOS command line monitor, and other network applications.

The MPC8245 instruction and data caches are enabled in copy-back mode for maximum performance. Includes the `dCacheFlush()` and `dCacheStore()` routines for cache management during interaction with DMA-driven peripherals and the `SyncCache()` routine for cache coherency after download of executable code.

Performs initialization and auto configuration of the A3000's PCI bus. When a device is found, the vendor ID, device ID, and device instance are announced to all registered drivers. Drivers respond using the `pciBaseAddr()`, `pciIsrCreate()`, `pciCfgRd()`, and `pciCfgWr()` calls to program the device. All devices found during auto configuration are listed in the TargetOS boot menu.

NVRAM driver stores configuration parameters such as startup mode, baud rate, and IP address in the A3000's flash memory. Configuration parameters have user-defined default values and can be modified using the TargetOS boot menu.

Periodic kernel tick driver supports the task sleeps and service call timeouts provided by TargetOS. The default tick interrupt frequency is 100 times per second.

Drivers for the MPC8245's four general-purpose timers allow applications to install callback functions that are invoked after high-resolution timing intervals, in either one-shot or periodic mode. User callback functions are installed as the timer interrupt handlers.

Flash program installer allows easy updates. Controlled via the TargetOS boot menu, the programmer downloads an ELF or S-record file via the serial port or Ethernet interface and installs it in the A3000's flash memory.

Flash file system driver supports TargetFFS™, Blunk Microsystems' high performance flash file system. TargetFFS provides a POSIX and Standard C compliant API and guaranteed file system integrity across unexpected shutdowns. ►

CONTACT INFORMATION

- Visit our web site:
www.blunkmicro.com
- Customer Support:
(408) 323-9833
- Technical Support:
(408) 323-1758
- Fax:
(408) 323-1757
- E-mail:
sales@blunkmicro.com
- Address:
**6576 Leyland Park Drive
San Jose, CA 95120
USA**

AVAILABLE COMPONENTS

TargetOS

Real-time, deterministic, multi-tasking, priority-based, preemptive kernel. Includes Standard C library.

TargetTCP

TCP/IP protocol stack with Berkeley Sockets API.

TargetFFS

POSIX compatible flash file system with wear-leveling and integrity across unexpected resets.

TargetLAPB

ISO/IEC 7776 protocol stack provides flow control and reliability on point-to-point networks.

Integrated with CodeWarrior™, the development environment from Metrowerks with an integrated source level debugger, compiler, assembler, linker, editor, and GUI make tool. Includes kernel awareness and event trace DLL.

Royalty free. Includes full ANSI C source code, default compiler settings, linker command files, user's manual, and one year of technical support. ■

Licensing Terms

Blunk Microsystems' A3000 BSP is royalty free. Purchasers are granted a non-exclusive license to use, modify, or extract the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.