



ANSI C code for building home networking and consumer electronics devices

Efficient Multithreading
4X more efficient than conventional architectures

Small Footprint

Add UPnP, Wi-Fi Protected Setup, LLTD with 120 kB of code

Modular

Microkernel

Target different features to different hardware with a mouse click

Standard APIs

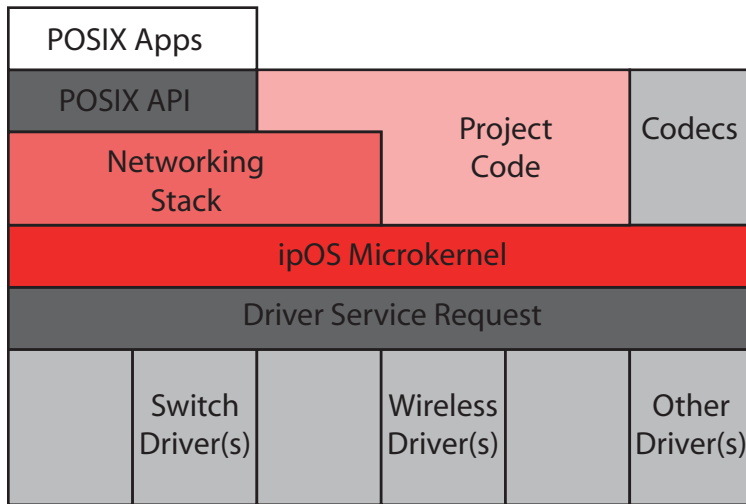
POSIX applications run natively

Automatic QoS

Identify, prioritize, and shape voice, video, games, and file sharing traffic

Tuning and Quality

Tools show what is consuming MIPS, memory, latency



Software Architecture

Multithreaded Software

The UbiCom Software Development Kit is used to build consumer networking devices and media players based on the UBICOM32 CPU architecture. The UbiCom platform uses multithreading, on-chip memory, and efficient software to achieve over four times the performance of a single threaded processor at the same clock frequency and die area.

Multithreaded hardware, such as the StreamEngine 5000 processor family, needs multithreaded software to achieve maximum performance. Multithreaded software operates more efficiently by avoiding cache misses and pipeline hazards. This results in a performance gain of over 100% over single threaded software.

Event Driven

The UbiCom microkernel does not perform software context-switches. Each thread runs only one stack. This saves memory that would normally go to maintaining several kernel and user software stacks.

The UbiCom SDK is inherently event-driven. The microkernel is based on a run-to-completion programming model. In this scheme, all work is handled with event loops. When an event occurs the system does all processing necessary to handle the event without performing a

Efficiency

100% performance gain over single threaded software

never perform a software context switch

software context switch. Context switching does no useful work and can consume roughly 10% of available CPU cycles in a traditional embedded system.

Control latency

control latency down to the microsecond in software

Each thread in the Uvicom platform runs an event loop for processing events. Events are sorted so that latency sensitive events are processed by one set of threads, while events that are less sensitive to latency are processed by another set. Deterministic multithreading and the run-to-completion programming model ensure that events can always be processed within a strict deadline.

Uvicom invented deterministic multithreading. Specific threads may execute on a fixed schedule. Hardware frequency clocks use this deterministic scheduling to precisely measure the latency and jitter of media streams.

A typical design on the IP5160 processor will use one thread per I/O port or peripheral, one thread to handle driver service requests, one for each codec, and one or more threads for mainline application code.

Zero-copy networking

Copying data is wasteful and should be avoided if possible. The Uvicom SDK employs a technique called zero copy networking. The SDK performs all key networking functions at the kernel level. In a conventional implementation, data is copied as it passed from kernel space to user space. Also, each task may also make unnecessary copies of the data as packets are processed. The Uvicom SDK does not copy packet data unless absolutely necessary. Pointers to packet data are passed between software modules within the networking stack without copying the underlying data.

avoid cache misses and unnecessary copying of data

Uvicom processors feature a significant amount of on-chip SRAM. The CPU core can operate on data within SRAM without going through caches and potentially suffering cache miss penalties. The developer can use software at design time or run time to decide whether data is placed in on-chip or off-chip memory. Under light loads, all data can be kept next to the CPU core for maximum performance. Under heavy loads, data that is manipulated often (such as packet headers) may be kept on-chip with seldom changed data stored off-chip. Together, on-chip memory and zero copy networking techniques make up the fast path. The developer has full discretion to tailor the fast path to the requirements of the project. The result is a very flexible and efficient implementation.

Automatic QoS

identify and prioritize a variety of media applications

The SDK includes software modules for identifying media streams on the network. Other modules are used to perform traffic shaping and analyze network characteristics.

Stream parsing

Many new networking protocols are text-based rather than binary and require parsing of XML data in various forms. The conventional way to handle XML data is to parse it using a DOM, or document object model, parser. The DOM reads in the entire XML string and represents this string in memory. This model is well suited to workstation programming but uses memory inefficiently. As the XML document gets larger, it consumes more memory.

The Ubicom SDK uses a stream parsing technique to handle XML, HTTP, and MIME data. The memory footprint of the stream parser is small and constant. The parser reads in text, and throws events when it reaches an element that can be parsed. All of this is done in a fixed heap size of roughly 200 bytes. The memory footprint for the stream parser is fixed, regardless of the size of the document being parsed.

parse HTTP, MIME, XML in about 200 bytes of fixed heap

Modularity

All of the features included in the Ubicom SDK are developed in a highly modular fashion. Any functions that would be duplicated in two places are broken out as a separate software module. Each module is documented with well defined APIs. This makes it very efficient to add new features. Often, the infrastructure needed to build a new feature is already available as a set of packages within the SDK. Sharing duplicate functions with modular software is the secret behind the small memory footprint of products built on the Ubicom platform.

over 260 documented packages make it easy to reuse code

Portability

The Ubicom SDK includes POSIX APIs to make porting software easier. The Ubicom platform can run POSIX applications efficiently because the overall system is multithreaded. Cache misses and pipeline hazards in a thread executing POSIX code are hidden because other threads will execute when a POSIX thread stalls.

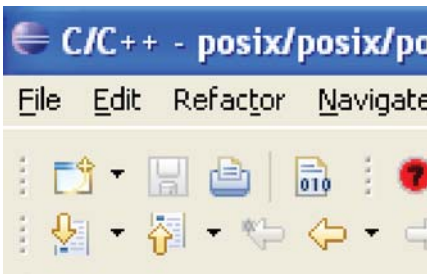
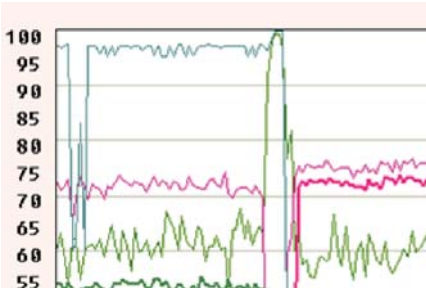
quickly port software using POSIX APIs

The Ubicom SDK includes the following POSIX APIs, with more being added in future releases:

- Red Hat newlib C and math libraries
- Pthread API
- IPv4 TCP and UDP sockets

Quality

target different features to different hardware with a mouse click



effective memory management

GNU Tools

The Uvicom SDK is based on the standard GNU tool chain including gcc 3.4.

Config Tool

The Config Tool is used to target different feature sets to different hardware designs using a single code base. The Config Tool has a graphical user interface that is used to select the features to be included in the project and the target board that will be used. The Config Tool calculates the software dependencies between all required packages and automatically generates all files necessary to build a project.

Profiler

The Profiler performs detailed performance analysis of the system in a non-intrusive manner. The Profiler measures system throughput for any interface, thread usage, how instructions are used, memory used per function, and end-to-end latency. The Profiler runs on its own thread on the CPU and send data to a PC through a programming dongle to a PC for viewing. Most of the I/O on the Uvicom platform is driven directly with software. This tool makes it easy to target specific ways to tune the performance of software.

Eclipse

A customized version of the Eclipse IDE (Europa version) is included in the SDK. Eclipse integrates the compiler, debugger, editor, CVS access, and Config Tool to maximize productivity.

Magic numbers

The Uvicom platform helps to ensure that heap and stack memory is not corrupted during use. Each type of object within the system with a magic number that is checked each time an access is made to an object. This technique is used extensively throughout the SDK.

Lock Deadlock Avoidance

A classic problem in multithreaded system design is circular waiting for resource. Two processes may both require data from each other in order to move forward, resulting in deadlock. The Uvicom environment enforces strict lock ordering as a way to avoid the lock deadlock problem.

Dump Analysis Tool

The Uvicom SDK includes a dump analysis tool. This tool provides detailed information for every memory block. It shows how packages use memory and can help track down memory leaks. The SDK also supports GNU GDB, so a corefile of all memory regions can be created for post failure analysis.

Heap Memory Management

The Ubicom SDK includes many functions that may be used to manage heap memory at run time. Memory tagging, guard bands, and block zeroing are useful in preventing and detecting memory leaks in a real system.

**prevent and quickly identify
memory leaks**

Stack Coloring

Stack coloring is used to determine the size of a stack at run-time. It is a technique for performance tuning and preventing stack overflows.

Type Reflection

Type reflection allows the system to determine the type of variables at run-time. This is useful for type conversion.

Part Number: IP5K-SLA-EWK

Site licenses for the Ubicom SDK are available. Please contact Ubicom sales for pricing and ordering information. The license includes one year of maintenance and rights to distribute binary software on Ubicom-based hardware platforms. Hardware reference designs and development boards are sold separately. Consult the Ubicom SDK release notes and datasheet for detailed features and specifications.

Ordering



IP5K-DPB-SDK-20

Ubicom develops multithreaded communications and media processor and software platforms that address the unique demands of real-time, interactive applications and multimedia content delivery in the digital home. The company provides optimized systems to OEMs targeting wireless routers, access points, VoIP gateways, streaming media devices, print servers and other network devices.

©2007 Ubicom, Inc. All rights reserved. Ubicom, StreamEngine, StreamEngine 5000, and ipOS are trademarks of Ubicom, Inc. All other trademarks are property of their respective owners. Features may change without notice.

**510 N Pastoria Avenue
Sunnyvale, CA 94085-3520
Phone: (408) 789-2200
Fax: (408) 739-2427
For US sales inquiries:
salesusa@ubicom.com**