

Treck TCP/IP

product information

High performance embedded systems are the key technologies fueling the innovative, high-growth applications of today's fast-growing markets. These include digital wireless, broadband access, digital audio, high-resolution imaging and digital motor control. A key reason for embedding pre-designed functions is to reduce the time it takes to get complex systems to market while speeding their proliferation.

Application Layer <i>User Applications</i>
Session Layer <i>BSD 4.4 Sockets</i>
Transport layer <i>TCP, UDP</i>
Network Layer <i>ARP, IP, ICMP, IGMP</i>
Data Link Layer <i>Ethernet, PPP</i>
Physical Layer <i>Network Interface Hardware</i>

The ability to rapidly design, test, debug and manufacture a device is crucial to the continued success of the electronics industry.

The embedded systems market is integrating computing and networking platforms capable of adapting to a

range of specific applications. Whether you are a silicon manufacturer or OEM, you need a top of the line networking stack to reduce your capital expenditure risk and meet the stringent requirements of the market.

Treck's networking protocols are specifically designed for embedded systems. We have created fast, efficient and reliable plug and play communication protocols so you can concentrate on your core competency and build designs that meet your "time to market" needs. Treck is an ideal TCP/IP stack for embedded systems designers.

Modular Design

The Treck TCP/IP protocol suite can be configured to only include certain modules. When a module is not utilized, it is not just deactivated, it is removed entirely to save valuable storage space in the often limited capacity of an embedded device.

True Zero Copy

Treck products are zero copy from the application all the way through the driver, including TCP, this increases processing speed.

Written Specifically for Embedded Systems

Treck is written from the ground up, not derived from publicly available Unix stacks. Berkeley 4.4 compatibility, small size and high performance are key elements of our design.

Can Run with or without RTOS

The stack is designed to easily integrate into an RTOS or Kernel. It can also run without an RTOS as its own task.

ROMable

Some program variables never change their value at run-time, but instead are assigned a hardcoded value at compile-time. These variables could be put into ROM, if the code was written in such a way to identify them to the compiler. We add the "const" keyword in front of such variables to indicate they never change their value so the compiler can locate them in ROM instead of in RAM. This conserves RAM.

Included protocols:

- TCP
- UDP
- IPv4
- ICMP
- ARP
- Ethernet
- SLIP
- RIPv2 Listener
- Ping Application

Optional Protocols

- PPP
- NAT
- IGMP
- DHCP
- FTP
- TFTP
- SNMP
- Telnet
- Mobile IP
- IPsec
- DNS Resolver
- HTTP
- POP3
- SMTP



Mobile



Network

Treck TCP/IP

product features

Highest Performance Available

- Zero copy to and from the application.
- Zero copy, even during IP fragmentation and reassembly
- Zero copy inside the stack (even for TCP)
- Zero copy to and from the device driver uses advanced features of device drivers, such as scattered data write (if supported by the device).
- Minimal RTOS/Kernel calls.
- Optimized for speed along the send/receive code paths.
- Extremely fast checksum computation.
- Treck Learn to allow the protocol stack to *learn* your performance requirements at run-time.

Designed for Embedded Systems

- Very small critical sections (usually fewer than 5 assembly instructions).
- Small code footprint (as small as 32K).
- Re-entrant and “Romable”.
- All of the RAM used by the stack is initialized by the stack.
- Easy to integrate RTOS/Kernel interface.
- Easy to use device driver API.
- Can run without an RTOS.
- Can run as its own task.
- Can run within the context of other tasks as a shared library.
- High bandwidth/high delay support (e.g. 32 bit window support) for set-top boxes.
- Multi-instance IP support
- ATM virtual channel support
- Full duplex TCP close
- Raw socket support
- Configurable packet trailer
- Device driver scattered send and receive
- User UDP scattered send and receive
- Configurable DHCP Client ID option
- Configurable DHCP initial state to INIT-REBOOT

Highest Quality TCP/IP Available

- Fully tested and qualified with over 300 different tests using many different compilers and processors.
- RFC2000 compliant.
- No porting required. You will never need to change our source code to integrate Treck TCP/IP into your environment.
- Everything can be set at run-time, including adding device drivers
- All default settings can be set at run-time and/or compile time.
- Fully ANSI “C” compliant.
- Warning-free Build.
- Can be built for size or speed.
- Single Header file for application code, interface code, and device drivers.

Supported Microprocessors

- Any 16, 32, or 64 bit microprocessor

Requirements to use Treck TCP/IP

- ANSI compliant “C” compiler
- Any CISC/RISC processor
- Sufficient ROM/FLASH for the protocol stack.
- Sufficient RAM for the maximum number of concurrent connections you will support.
- A network device
- A timer to signal the protocol.

For more information, please visit www.treck.com



Mobile



Network

Treck Inc.